Solid-State Circuit Breaker Component Simulation

Matthew Hughes
Marquette University

SOLID-STATE CIRCUIT BREAKER COMPONENT SIMULATION

by

Matthew Hughes, B.S.

A Thesis Submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for the Degree of
Master of Science

Milwaukee, Wisconsin

May 2021

ABSTRACT

SOLID-STATE CIRCUIT BREAKER COMPONENT SIMULATION

Matthew Hughes, B.S.

Marquette University, 2021

The landscape of direct current (DC) solid-state circuit breakers (SSCBs) is growing and so too is the choice of primary components. Early on in the development of electrical technology, DC electrical distribution networks were handicapped by short transmission distances, poor network configurations, and overall lowered efficiency. These issues arose out of the relatively lower voltage transmission capabilities, which do not exist in the modern day of semiconductors. There now exists a wide range of devices capable of withstanding voltage levels conducive to electrical transmission; though steady state efficiency of these devices remains a concern. Choosing primary components out of the growing selection pool can be cumbersome and prone to biased decision-making. Put forth in this thesis is a multifaceted methodology for determining primary components of SSCBs removing unwanted biasing, thus alleviating burden from the design process.

This methodology consists of accurately and consistently simulating efficiency of solid-state devices of a variety of types, topologies, and materials; as well as developing methods for the comparison of solid-state devices and the associated voltage suppression components. These comparisons do not rely on efficiency alone, but also account for thermal characteristics and costing; allowing for a mathematically based bespoke solution for specific design applications. These individualized solutions are additionally simulated within an experimental test fixture, using non-ideal components to produce further confidence in SSCB primary component choice.

As previously stated, the field of SSCB technology is growing and focusing solely on one aspect of SSCB performance may introduce complications later in the system design process. The purpose of using a multifaceted mathematical approach to decision-making is to reduce these complications. The results contained within this thesis show a more holistic approach to SSCB primary component choice is achievable and warranted.

ACKNOWLEDGEMENTS

Matthew Hughes

I would like to thank Dr. Nathan Weise for serving as Chair of my thesis committee and for his guidance, support and endless encouragement during my graduate studies. I would also like to thank Dr. Ayman EL-Rafaie and Dr. Joshua Carl for their support as members of my committee. Additionally I would like Dr. Joshua Carl for his support during my undergraduate studies. Without his encouragement I very well could have never attended graduate school. He changed my future from a "never was" to a "will be".

I would like to acknowledge Advanced Research Projects Agency–Energy for providing support for my research, without it, none of this would be possible. Additionally, I would like to acknowledge Marquette University, Opus College of Engineering, Electrical Engineering and Computer Engineering Department and Graduate School staff. All of who are humble, helpful, and wonderful to work with; thank you for providing an unlimited flow of knowledge.

Finally, I would like to thank my colleagues, friends and family for providing me strength through their continuous support and love.

DEDICATION

To my grandfather, Wendell.
Thank you for showing me that strength does not have to be loud, boastful or mean. Thank you for showing me that strength of will is more powerful than strength of body. Above all, thank you for showing me what it means to serve our country.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SOFTWARE

CHAPTER 1
INTRODUCTION

Electrical power transmission across the globe uses alternating current (AC) voltage systems, but this was not always the case. At the start of the electrical revolution, direct current (DC) voltage systems were more prevalent, but fell out of favor due to efficiency limits of the time. In more recent years, technology has progressed and those previous limits have been exceeded. As it stands, the current state-of-the-art DC circuit breakers do not match the same performance metrics as AC circuit breakers. This discrepancy will need to be remedied prior to any widespread installation of a DC voltage power transmission network. My effort to lessen the gap will be to provide context to why switching to a DC voltage power distribution network is desirable, identify the primary design concerns with DC circuit breakers, and provide a solution to validate traditional solid-state circuit breaker (SSCB) components.

1.1 Motivation of Research

In the late 19^{th} century a war was waged between great minds in the United States of America (US), that would later become know as the, "War of the Currents"[1]. Thomas Edison established the Edison Illuminating Company on the basis of DC technology, which is an effect of electrons traveling continually in a single direction. The resultant voltage never changes polarity. George Westinghouse on the other hand, promoted the use of AC, also known as "The Westinghouse Alternating Current System of Electrical Distribution", within his own enterprise the Westinghouse Electric Company [2]. The difference to DC technology being that AC technology uses expanding and collapsing magnetic fields to induce a voltage that continually changes polarity. The resultant current, rate of charge flow, changes direction with the change in voltage or potential energy per unit charge [3].

Though DC technology was widely adopted prior to the development of AC, it would succumb to the efficiency that AC technology would later provide. This efficiency was showcased at the 1893 Colombian World's Fair in Chicago, where Westinghouse Electric Company bid $5.25 per bulb as compared to the Edison backed General Electric's DC bid of $13-18 per bulb [4]. The cost savings provided by AC technology was a result of

a more efficient power distribution system. This efficiency was gained by transmitting power at relatively higher voltage levels.

In modern terms, voltage levels are divided into ranges including: ultra-low voltage (ULV), low-voltage (LV), medium-voltage (MV), high-voltage (HV), and ultra-high voltage (UHV). The specific voltage value depends on voltage type, use case, and purpose. For instance, LV in a three-phase AC transformer used in a transmission system is considered to be less than $15kV$ [5]. When determining an arc-flash safety boundary of switch gear, above $15kV$ would be considered HV [6]. These levels are determined by many governing bodies including: International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE), National Electrical Manufacturers Association (NEMA), and the National Fire Protection Association (NFPA) [7]. The voltage levels to be used for the purposes of this thesis are listed in Table 1.1.

Range	**AC Voltage (VRMS)**	**DC Voltage (V)**
UHV	≥34,500	>500,000
HV	15,000 to <34,500	9,000 to 500,000
MV	1,000 to <15,000	1,000 to <9,000
LV	50 to <1,000	50 to <1,000
ULV	<50	<50

Table 1.1: Voltage Level Ranges

Edison Illuminating Company's early power distribution was LV DC, which limited utility service to no greater than one mile from any generation point or power plant. This limit was due to efficiency losses and infrastructure costs. George Westinghouse on the other hand would develop a MV AC distribution system, with the help of Nikola Tesla, and install hydro-electric AC turbine-generators at Niagara Falls, New York and then transmit the generated power 20 miles to Buffalo, New York.

The relatively long distance transmission achieved by the Westinghouse Electric Company was made possible by the use of transformers. A transformer would be used to raise the voltage level for power transmission from the source to a local distribution

center. The distribution center in turn steps down the voltage for short duration transmission and then if necessary voltage is stepped down further at the termination point. An example diagram of this process can be seen in Figure 1.1.

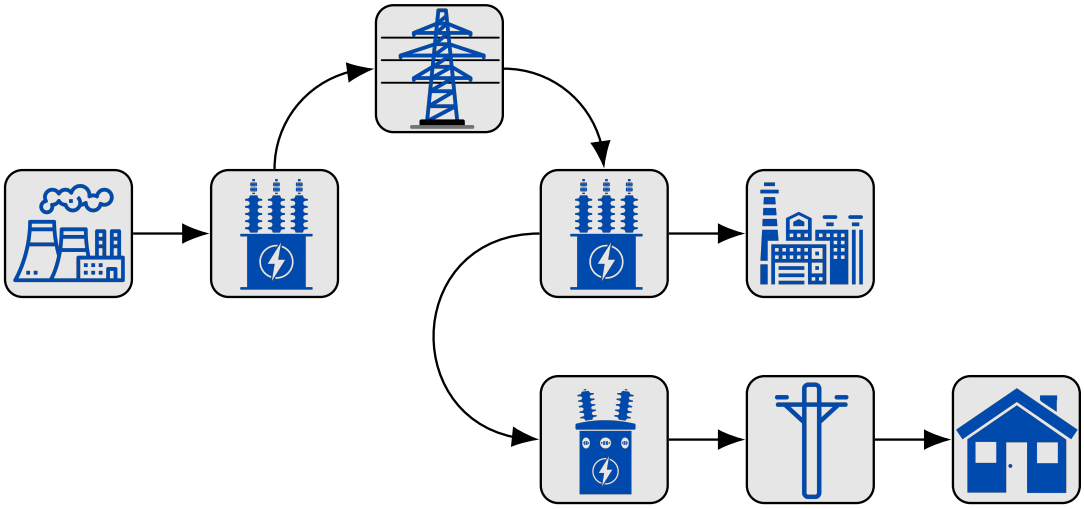


Figure 1.1: Example Transmission Diagram

How raising the voltage during power transmission results in better efficiency can be explained using the efficiency relationship, the power law, and Ohm's Law. In the efficiency relationship, Equation 1.1, the amount of losses,(P_{losses}, $watt$) is subtracted from the supplied power, (P_{supply}, $watt$), and directly affects the transmission efficiency, $efficiency$, %. This loss in power, (P, $watt$), can be related to the transmission voltage, (V, $volt$), and current, (I, $ampere$), using the power law, as seen in Equation 1.3. Furthermore, the relationship between resistance, (R, ohm, Ω), and the transmission voltage and current can be determined using Ohm's Law, as seen in Equation 1.2.

$$efficiency = \left(\frac{P_{supply} - P_{losses}}{P_{supply}} \right) \times 100 \tag{1.1}$$

$$V = I\,R \tag{1.2}$$

$$P = I\,V = I^2\,R = \frac{V^2}{R} \tag{1.3}$$

These power losses are also referred to as line losses or "I^2R" losses and are the result of the resistance inherent to the material used in the current carrying conductor [8].

This material is commonly copper for small scale transformers or aluminum for large scale, due to cost considerations [9] [10]. In suspended transmission lines aluminum is used due to weight considerations. All things being equal, an assumption that the inherent resistance of a transmission line is constant can be made. Therefore, the transmission voltage or current must be changed to reduce the power losses. From Equation 1.3, we can determine that reducing current has a much larger affect of reducing power losses than reducing voltage. This is due to the inverse relationship current has with resistance.

Simply reducing the transmission current would also reduce the transmitted power, this is undesirable. To maintain the transmitted power while reducing the current, voltage must be raised. As previously mentioned, this is completed using a transformer. Referring to a simple transformer model, Figure 1.2, it can be seen how energy would be transferred from the primary side (1) to the secondary side (2) of a transformer while maintaining an ideal balance of power, $Power\ In = Power\ Out$. The relationship of the power transfer is further explained using the turns ratio, Equation 1.4, where the inverse relationship of voltage (v) and current (i) across the transformer can be seen as well as the affect of the number of turns (N) [11].

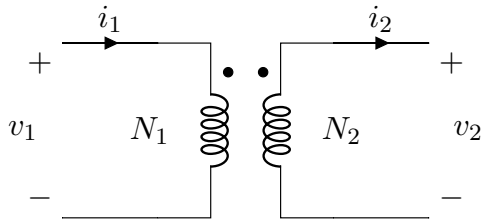


Figure 1.2: Transformer Model

$$\frac{v_1}{v_2} = \frac{N_1}{N_2} = \frac{i_2}{i_1} \tag{1.4}$$

A simple transformer construction would be two coils of a current carrying conductor, such as copper wire, wrapped around a single core of any material. Ferrous materials, such as an iron core, provide a higher efficiency of coupling and tend to be more desirable. In order for the transformer to function properly the orientation of the coils

must be in a way to invoke Faraday's law of induction. The proper orientation enables the transfer electrical energy between the two coils [3]. This can be seen in Equation 1.5, given a loop of wire in a magnetic field, the magnetic flux (Φ_B, $Weber$) is the surface integral of the vector dot product between the magnetic field (B, $Tesla$) and a portion of the surface area (dA, $meter^2$). This integral is defined for any surface ($\Sigma(t)$) bounded by the moving wire loop [12].

$$\Phi_B = \iint_{\Sigma(t)} B(t) \cdot dA \tag{1.5}$$

In the case of AC, the wire loop is not physically moving to intersect the magnetic field, but the magnetic field is expanding and collapsing intersecting the wire loop. This is why DC can not use transformers to raise or lower voltage, the generated magnetic field is static. Again, the property of Faraday's law of induction in conjunction with the turns ratio can be used to raise or lower voltage as necessary while maintaining the magnitude of power transferred. The inverse relationship of voltage and current across a transformer makes this possible. Raising voltage on the secondary side of the transformer, the output, is advantageous to transferring power due to the reduction of current on the secondary and therefore the reduction of line losses.

As a result of the efficiency gains in power distribution the US national power grid was designed and built to support an AC system. Therefore infrastructure, buildings, and devices were designed and built to operate on AC voltage. Though this dependency on AC voltage has been decreasing since the rise of semi-conductors. In recent years there has been a gain in the desire to operate devices on DC voltage which in turn is driving a need for DC voltage infrastructure and power distribution. These devices range from consumer products such as personal electronic devices, light-emitting diode (LED) lighting, and electric vehicles (EVs) to large industry computer data centers and server farms that make up the backbone of the internet as well as companies like Amazon, Apple, Google, and Microsoft.

In addition to devices moving predominately to DC voltage, sources of power are also becoming more DC voltage dependent [13]; one such area is microgrid power generation. The U.S. Department of Energy (DOE) defines a microgrid as "a group of

interconnected loads and distributed energy resources within clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid. A microgrid can connect and disconnect from the grid to enable it to operate in both grid-connected or islandmode." [14]. Microgrids are desirable due to improved reliability and power grid security as well as minimizing energy costs to the consumer [15] [16]. Microgrids also tend to be powered by renewable energy sources, such as wind and solar, but these sources provide power intermittently and require energy storage to ensure a reliable supply of power [17].

Just as with power generation, energy storage is also becoming more DC voltage dependent [13]. Overall, DC voltage energy storage is predominately in the form of battery systems. These battery energy storage systems are naturally advantageous to solar photo-voltaic (PV) systems which are also a DC energy supply as well, making compatibility between generation and storage less of a problem [18].

As stated earlier, microgrids can operate in both grid-connected or islandmode. When grid-connected the AC voltage must be rectified to DC voltage in order to supply the microgrid and vice versa. If the power transmission grid was DC based, this rectification would be unnecessary. Though, an interconnection device between systems of different voltage levels would be required. Such devices called high voltage direct current (HVDC) Back-to-Back Stations are already in use in the US national power grid.

These HVDC Back-to-Back Stations are used to interconnect between major electric grids, due to the mismatch of frequency between the electric grids. Interconnecting between AC voltage systems must be synchronous, otherwise the proper transfer of power through the interconnect will not occur. Voltage between AC systems is considered synchronous when the frequency of each system is matched. Synchronicity is not necessary in a DC voltage system, this affords the benefit of allowing asynchronous connections of power sources [19] [20]. An example of one of these HVDC Back-to-Back Stations is depicted in Figure 1.3; where it can be seen that power converters are co-located allowing for the interconnection of asynchronous AC systems.

It is desirable to have a DC voltage system be the interconnection between multiple microgrids as well as the interconnection between conventional primary sources of

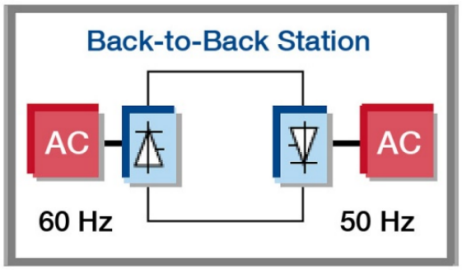


Figure 1.3: HVDC Back-to-Back Station Configuration [21]

power generation, such as fossil fuel power plants [17]. A prime example of a HVDC interconnect being used is between the Electric Reliability Council of Texas (ERCOT) and the Eastern Interconnection of the North American Electric Reliability Corporation (NERC), these interconnects are also known as DC bus ties [22]. These type of HVDC Back-to-Back Station interconnects are ubiquitous in electrical power transmission. A map provided by National Renewable Energy Laboratory (NREL) of these interconnects in North America can be seen in Figure 1.4.

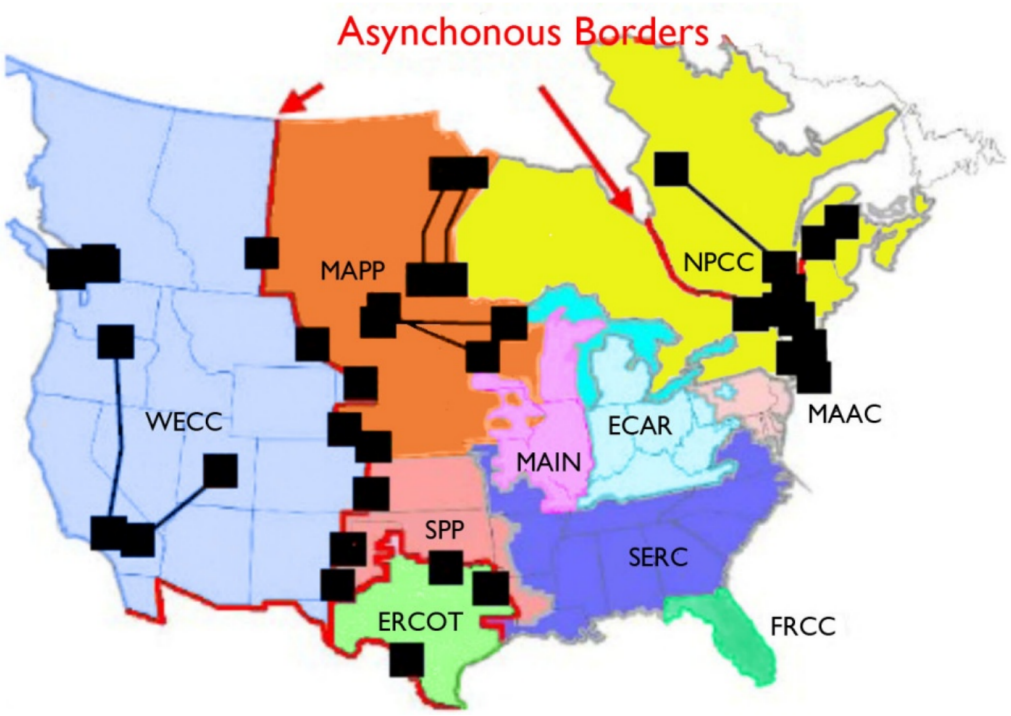


Figure 1.4: North American HVDC Back-to-Back Station Locations [21]

Along with asynchronous operations, DC power transmission can be more cost effective than AC power transmission [19] [20]. As seen in Figure 1.5, there exists a break-even distance where the investment cost of DC power transmission is equal to the investment cost of AC power transmission. Generally speaking the break-even point happens at a transmission distance of few hundred miles. Along with transmission distance, this cost is relative to the transmission voltage level as well. The higher the transmission voltage level the larger the investment cost, but the closer the break-even distance is to the origin. The steep increases of investment cost of an AC system comes from the required compensation of series and shunt reactive power, depicted in Figure 1.5 as series and shunt compensation (SSC). This type of compensation is only required in AC power transmission.

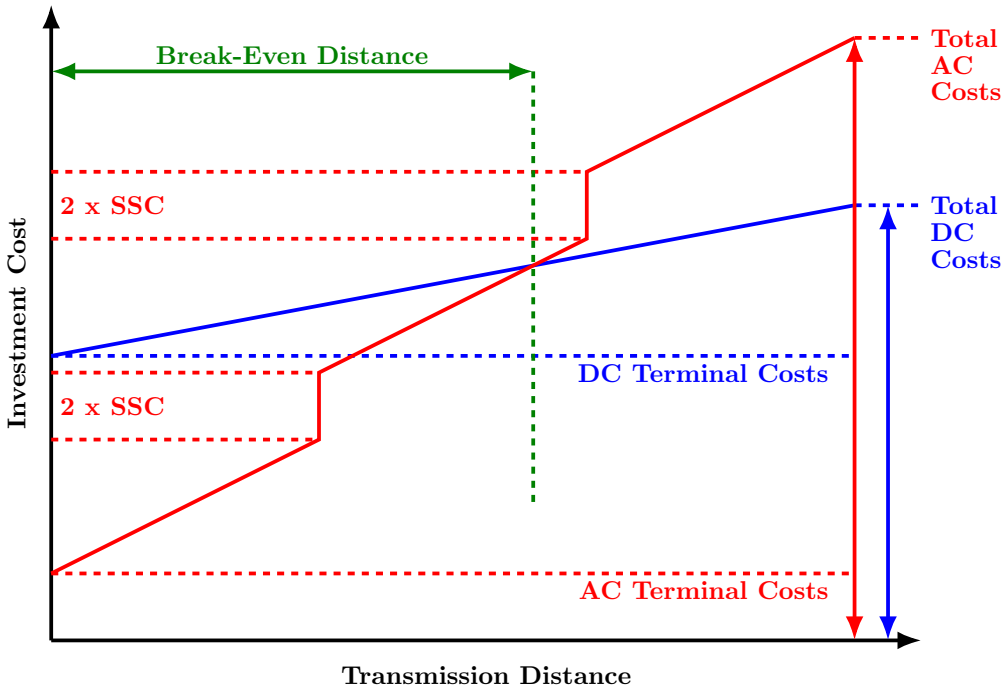


Figure 1.5: Cost Comparison of AC versus DC transmission [23]

Again, at very high voltages and long distances DC transmission is more cost effective than AC transmission. This is due to the lack of reactive power generated in DC transmission. Reactive power is the electric power exchanged between the magnetic field of an inductor or the electric field of a capacitor and the driving source [3]. Reactive

power is never converted into non-electric power, therefore it does no work. Since reactive power is the consumption of energy without producing work, any addition of reactive power to a system results in a reduction of efficiency.

Overcoming these induced reactances to produce enough active power at the load, requires reactive power compensation [24]. Active power, also known as real power or average power, is the power converted from an electric form to a non-electric form and vice versa [3]. Real power consumes energy, but produces work and has the units of watts (W). In AC voltage systems the requirement of reactive power compensation increases as voltage increases or transmission distance increases or as both increase. Extra equipment and infrastructure is required to be installed to offset the reactive power produced. Because a DC voltage system would not require this compensation, this alone makes a DC voltage system more desirable from an equipment cost standpoint.

The relationship of a reactive power to efficiency can also be depicted using a power triangle [3]. In Figure 1.6, three sides of a right triangle are the reactive power, (Q), average power, (P), and apparent power, ($|S|$) [3]. As stated earlier reactive power does no work and is always aligned to the imaginary axis. Also, reactive power has the units of volt-amp reactive (VAR) to differentiate it from average power. Again, average power can be converted into non-electric power, therefore it can do work and is aligned to the real axis. Apparent power has the units of volt-amps (VA) and represents the necessary capacity to supply the active power [3]. In other words, apparent power is the total power required to supply the average power and overcome the reactive power.

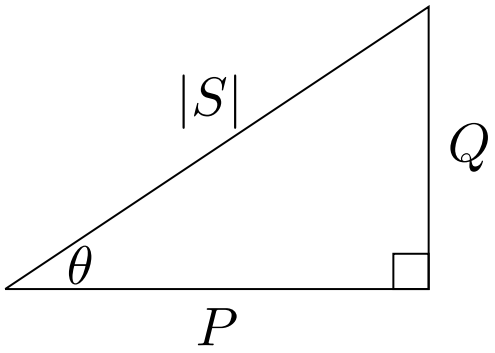


Figure 1.6: Power Triangle

Another representation of efficiency can be made using the angle between apparent power and average power, annotated as θ in Figure 1.6. Trigonometric rules can be applied to the power triangle to define the power factor (pf), as shown in Equation 1.6 [3]. The smaller the angle between these two powers, the closer the power factor is to being equal to one, which is the ideal efficiency of 100%.

$$pf = \frac{P}{|S|} = cos(\theta) \tag{1.6}$$

A prime example of how reactive power can be introduced in an AC voltage system and drive down efficiency is via a power transmission line. An AC transmission line can be represented in a series of resistances (R), inductances (L), and shunt capacitances (C), as seen in Figure 1.7 [25]. The resistance in an AC transmission line is greater than the resistance in a DC transmission line [26]. Also, the inductance and capacitance in an AC transmission line increase with transmission distance.

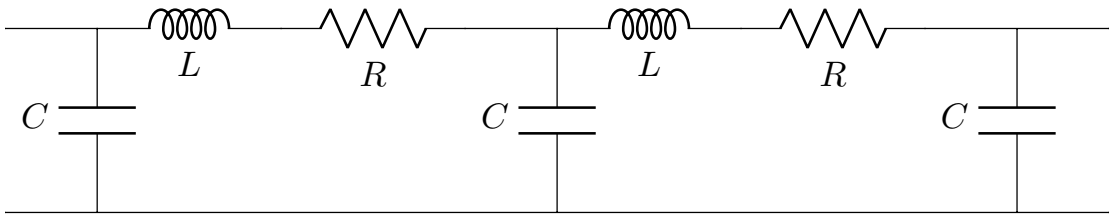


Figure 1.7: AC Transmission Line Representation

The increase of resistance in an AC transmission line is due to the skin effect, proximity effect, hysteresis and eddy currents [26]. The current distribution across the cross-sectional area of a conductor carrying AC is not uniform [26]. This phenomenon is known as the skin effect; which can also be characterized as the reduction of current density in the center of a conductor carrying AC [27]. This loss of current density essentially reduces the size of the conductor and therefore increases the effective resistance of the conductor at higher frequencies.

A continuation of the non-uniformity of current flow in an AC transmission line can be attributed to the proximity effect. This effect occurs when conductors carrying AC are in close enough proximity that their mutual induction affects the current distribution

in the adjacent conductors [26]. This effect causes an increase in the resistance that is inversely proportional to the distance between the conductors [26].

Efficiency losses from hysteresis and eddy currents are a function of the current flowing through the conductor and increase the effective resistance [26]. These losses are more prominent in steel reinforced aluminum conductors (ASCR) and at high currents could experience an increase of resistance by more than twenty percent [26].

Again, the inductance in an AC transmission line increases with transmission distance and contributes to average power losses. This can be seen using the inductance of a magnetic circuit (L, $Henry$), as defined by the number of flux linkages (λ, $Weber - turns$) per current (I, $ampere$), as seen in Equation 1.7 [26]. This equation does not explicitly show the inductance per unit length of an AC transmission line, but can be better defined in Equation 1.8 [26]. Here the inductances (L_a, L_b, L_c) per unit length per phase are represented in a three-phase transmission line with conductors of equal distance apart; where the radius of conductor is D_{aa} (m) with a separation of transmission lines of distance D (m). This results in inductance per unit length with units of Henrys (H) per meter (m).

$$L = \frac{\lambda}{I} \, [Wb/A] \tag{1.7}$$

$$L_a = L_b = L_c = 2 \times 10^{-7} \, ln \left(\frac{D}{D_{aa}} \right) [H/m] \tag{1.8}$$

Just as with the inductance, the capacitance in an AC transmission line increases with transmission distance. Knowing the capacitance (C, $Farad$) of long parallel conductors can be considered conducting plates of a capacitor they can be defined using Equation 1.9 [26]. In Equation 1.9, the voltage (V, $volt$) present in the transmission line is separated by a dielectric material (air). This in turn will cause a charge (q, $Coulomb$) of equal magnitude to develop between adjacent AC transmission lines as well as between the AC transmission line and ground. Much the same as in the case of line inductance, the relationship to distance in Equation 1.9 is not inherently obvious and can be better defined in Equation 1.10 [26]. Here the capacitances (C_{an}, C_{bn}, C_{cn}) of the positive-sequence per unit length capactiance to neutral are represented in a three-phase transmission line with conductors of equal distance apart; where the permittivity of free space (ϵ) is a known

quantity, with all conductors having a radius r (m), and be equally spaced a distance of D (m). This results in capacitance per unit length with units of Farads (F) per meter (m).

$$C = \frac{q}{V}\,[F] \tag{1.9}$$

$$C_{an} = C_{bn} = C_{cn} = \frac{2\,\pi\,\epsilon}{ln\left(\frac{D}{r}\right)}\,[F/m] \tag{1.10}$$

As demonstrated, inductance and capacitance is introduced into an AC transmission line as power transmission distance is increased. This increase in resistance, inductance and capacitance also increases impedance; which reduces power transmission efficiency.

To further explain, impedance is the total opposition to an alternating current. The relationship between impedance (Z, ohm, Ω), voltage (V, $volt$), and current (I, $ampere$) can be seen in Equation 1.11 [3]. Impedance is any combination of resistance, inductance, or capacitance inherit to the circuit components used and their orientation relative to each other. Where as, resistance is the capacity of materials to impede the flow of current, (R, ohm, Ω) [3]. Inductance relates the induced voltage from a time-varying magnetic field to the current, (L, $Henrys$) [3]. Capacitance relates the displacement current from a time-varying electric field to the voltage, (C, $farads$) [3]. Therefore, impedance in the frequency domain corresponds to the resistance, inductance, and capacitance in the time domain. Furthermore, impedance inherently has real and imaginary components, being resistance and reactance respectively. These properties are further defined in Table 1.2 [3].

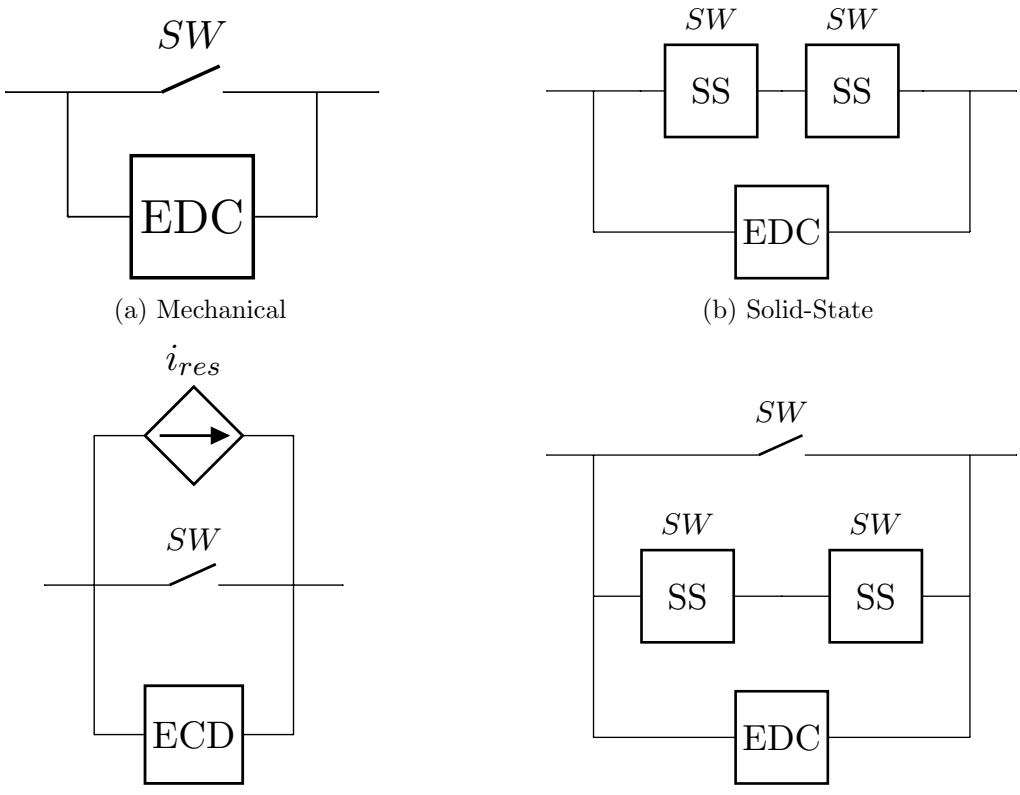
$$V = Z\,I \tag{1.11}$$

Element	**Impedance**	**Reactance**
Resistor	R	-
Inductor	jωL	ωL
Capacitor	j(-1/ωC)	-1/ωC

Table 1.2: Impedance and Reactance Values

Each one of the factors previously stated demonstrate how DC transmission can be more efficient and more cost effective than AC transmission at high voltages and long distances. This fact is further reiterated in the previously mentioned Figure 1.5. This alone displays why a concerted effort should be made to move towards HVDC power transmission [28]. Though, this transition should not come at the cost of safely operating circuit breakers.

When opened under load circuit breakers experience electrical, thermal, and dynamic stresses. These stresses are present regardless if the circuit breaker is installed into an AC or DC voltage system. Furthermore, these stresses are present in all circuit breaker types including: mechanical, solid-state, hybrid mechanical resonant, and hybrid mechanical solid-state designs as depicted in Figure 1.8a, Figure 1.8b, Figure 1.8c, Figure 1.8d respectively. These circuit breaker types and their attributes will be covered in more detail within, "Chapter 2.5: Review of Literature", of this thesis.



(a) Mechanical

(b) Solid-State

(c) Hybrid Mechanical Resonant

(d) Hybrid Mechanical Solid-State

Figure 1.8: Circuit Breaker Design Types

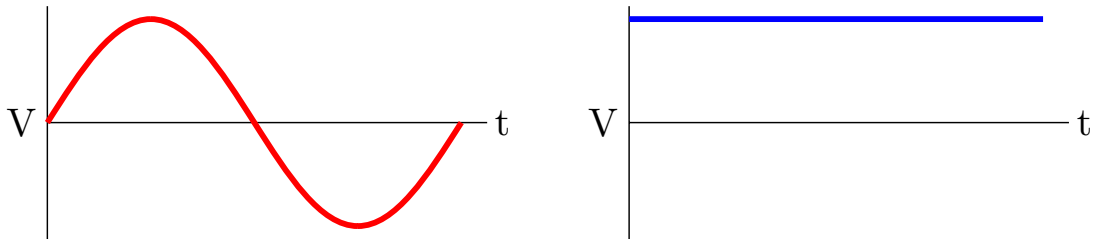
Each of these circuit breaker types have at least a main conduction path and an energy dissipation path. The purpose of the main conduction path is to transmit energy from the input of the circuit breaker to the output of the circuit breaker. When that path is broken and the flow of energy is suddenly halted any excess energy will remain. The purpose of the energy dissipation path is to intake this excess energy using an energy dissipation component or configuration of components and methods in the form of a energy dissipation circuit (EDC). Without the energy dissipation path the main conduction device would experience all the stresses that were created when the flow of energy was suddenly halted.

In all of the previously mentioned circuit breaker design types, with the exception of SSCBs, the electrical and thermal stresses experienced when a circuit breaker is opened under load manifest in the form of an electrical arc or arc discharge.

An electrical arc is the breakdown of a gas electrically that results in the production of a prolonged electric discharge [29]. That electric discharge reduces the working life of the circuit breaker and can result in damage to the circuit breaker and its associated infrastructure. Along with damaging equipment, electrical arcs have the potential to harm personnel, if not appropriately addressed using some form of arc suppression.

Arc suppression is the reduction of electric discharges resulting from opening electrical connections while power is applied. The act of arc suppression can be completed in many ways depending on the type of circuit breaker. Though, most circuit breaker designs employ more than one method of arc suppression.

It should be noted, that the more energy present when a circuit breaker is opened, the larger the electrical arc created. This is true in regards to DC as with AC. Though in AC there exists a natural zero crossing, as seen in Figure 1.9a. This property of AC quickly extinguishes any electrical arc created between the separating conductors inside a mechanical circuit breaker. In DC this zero crossing does not exist, as seen in Figure 1.9b, and the electrical arc is sustained for several moments after opening. This creates a barrier for HVDC to be adopted in the market, due to a higher level of sophistication of design needs and costs [30].



(a) Alternating Current (b) Direct Current

Figure 1.9: Illustration of Zero Crossings in Voltage Types

In review, DC voltage transmission is more cost effective at higher voltage levels and at longer distances than AC voltage transmission. There is also a growing demand for DC voltage power distribution networks due to the compatibility with renewable energy power sources and energy storage. Though there still exists a technology gap for DC voltage systems in the area of isolation equipment longevity, efficiency, and safe operation due to electrical arc discharges. Therefore, the development of HVDC circuit breakers are an essential key in the commercialization of HVDC systems [31].

As previously mentioned there are multiple solutions for DC circuit breakers. The following research will focus on a SSCB design. Specifically, the performance of primary components in conduction and energy dissipation. As a reminder, with a SSCB solution no electrical arc discharge occurs during the normal operation. The design of a SSCB has no moving parts, specifically no separation of current carrying conductors. Therefore, there would be no stress from an electrical arc discharge, because there is no arc formed to be discharged. This does not mean a SSCB operates without stressing forces. Other stresses to the system due occur, most importantly with energy disbursement. Though it is important to note these stresses are not unique to SSCBs and exist in all circuit breaker designs.

Prior to any analysis a thorough review of circuit breaker design history will be completed. In this review the advantages and disadvantages of the previously mentioned circuit breaker types will be discussed. Most importantly, the review of recent literature and research of SSCB designs will be completed. After which the analysis and simulation of SSCB primary components will be completed. The following sections contain an overview of the research that will be covered in this thesis.

1.2 Scope of Thesis

The following analysis will focus on fully developing the theoretical and analytic relationships necessary to make informed decisions on the primary components used in the design of a SSCB. The primary components under consideration will be the main conduction switch and the energy dissipation device. The initial theoretical analysis will be expanded to a set of selection criteria that will output a set of potential candidates for SSCB components. Inputs to the criteria will be the following application parameters: bus voltage not to exceed 1.2 $kVDC$, resistive load not to exceed 100 Ω or fall below 30Ω, inductive load not to exceed 100 μH, and various operating temperatures between -55$^{\circ}C$ and 175$^{\circ}C$.

During the simulation phase of research, various tools will be employed to generate data from the theoretical analysis for the selection criteria; including but not limited to Matlab and LTspice. The following component attributes will be used in the selection criteria: steady state efficiency, thermal load, cost, transient energy dissipation, as well as other metrics.

The results from the simulation phase will be evaluated and scored. The evaluation will result in appropriate primary component choice for the chosen SSCB design. This evaluation process will take into account each of the prior mentioned metrics. If deemed appropriate these metrics will be weighted as per importance.

Once primary components have been chosen, the process of generating experimental results can move forward. Though this thesis will not cover any such results, a precursor to any experimental work would be the test fixture design. A theoretical basis for a test fixture design will be covered. Included with the design considerations will be a limited simulated implementation.

At the completion of this thesis a review of state-of-the-art SSCBs will have been completed, a reasonably thorough investigation of design methods conducted, an implementation of those methods, and generation of simulation results verifying or refuting those methods. As well as the development of a test fixture design process.

1.3 Organization of Thesis

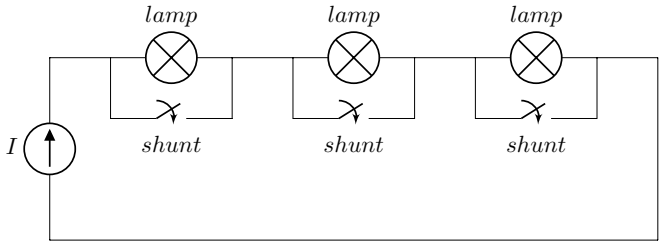
This thesis began by providing historical context of AC voltage and DC voltage power transmission systems in the US. Included in that history was why there exists a need for safe DC voltage circuit breakers and a plan on how to achieve that safety using SSCB designs. In Chapter 2.5, a comprehensive review of circuit breaker design history will be conducted, with focus on SSCBs. Included in this review will be current state-of-the-art SSCB design solutions. In Chapter 3.4.4.5, a mathematical analysis will be completed in order to derive appropriate component metrics. As well as the introduction of any components deemed necessary from the analysis. In Chapter 4.5.3, the proposed component metrics will be used to generate data through various software simulation programs creating a baseline for decision-making. In Chapter 5.2.4, the results from the analysis and simulations will be compared to verify design and component attributes. In Chapter 6.6, the conclusions reached from the results data will be presented and contain all pertinent decisions and recommendations on future work. Finally, in Chapter 7.2.12.3 a test fixture design will be discussed; including any necessary analysis and recommended conclusions.

CHAPTER 2
REVIEW OF LITERATURE

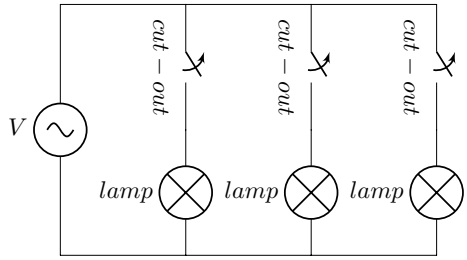
The following comprehensive review of current design strategies for SSCBs will commence by reviewing early automatic device designs leading up to modern day designs. The main focus of the review will be the steady-state efficiency and transient operations of each of the previously mentioned DC circuit breaker designs.

2.1 Early Automatic Devices

Early electrical lighting systems developed by the Edison Illuminating Company using DC systems would redirect or shunt current around faulty components instead of interrupting current [32]. This was due to the lighting systems being installed in a series path connection with a current dependant supply. Conversely, the Westinghouse Electrical Company would use a voltage dependant AC supply [33]. This allowed AC systems to more easily utilize both series and parallel path connections, but would not benefit from shunt devices. Therefore, an early form of interrupt devices were developed, known as electrical cut-out devices. Example electrical system layouts of these series and parallel paths are seen in Figure 2.1a and Figure 2.1b, respectively.



(a) Series Circuit



(b) Parallel Circuit

Figure 2.1: Electrical System Layout Depictions

An early US patent of a shunt circuit, similar to the devices depicted in Figure 2.1a, was filed on October 20, 1885 by Thomas A. Edison of Menlo Park, New Jersey. A portion of the filed schematic can be seen in Figure 2.2, with the full schematic available in Appendix B: Schematics. This design would reestablish current to the downstream devices in the event an open circuit occurred, but would not protect against an abnormally high current flow. In patent US328573, Edison explains the device operation, "The parts will be constructed and adjusted so that the magnet will attract the armature and keep the shunt open when there is any current in the magnet, and will not permit the shunt to close or remain closed except upon a total cessation of current in the magnet coils." [32]

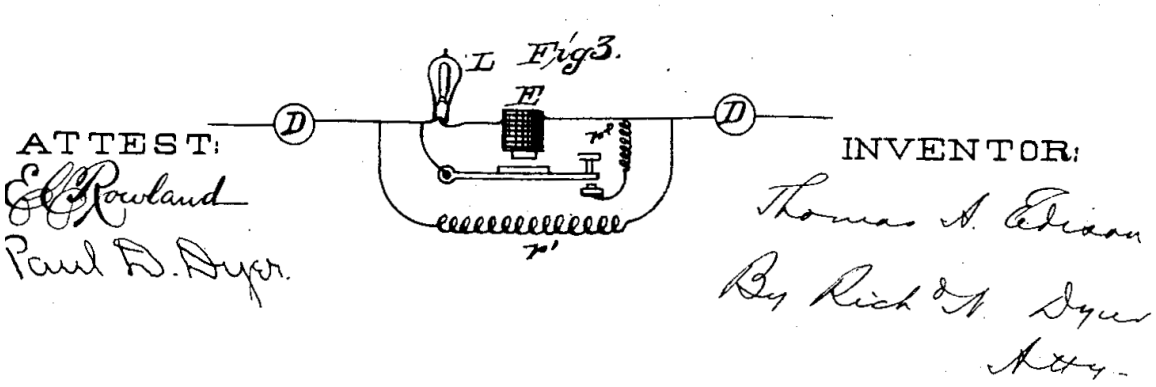


Figure 2.2: System Of Electric Lighting Patent [32]

An early US patent of an electric cut-out, similar to the devices depicted in Figure 2.1b, was filed on November 14, 1893 by Elihu Thomson of Swapscott, Massachusetts. A portion of the filed schematic can be seen in Figure 2.3, with a full schematic available in Appendix B: Schematics. This design would automatically break the electrical connection if current were to become abnormally high and extinguish the resultant arc in a non-conductive fluid. In patent US508652, Thomson explains the cut-off condition, "My present invention relates to electric cut-offs for cutting off or breaking an electric circuit on occurrence of an abnormal current therein." [34] Thomson goes on to further explain the method arc suppression, "One of the features of my invention is that the interruption of the circuits accomplished by drawing open the contact pieces and carrying one or both of them down into a body of oil which extinguishes the arc." [34]

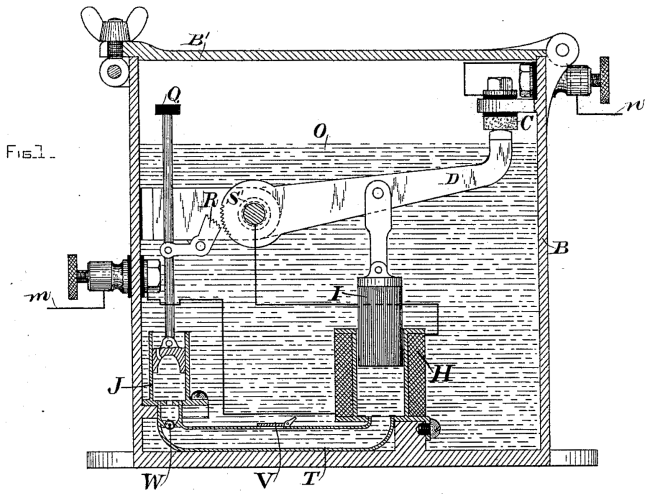


Figure 2.3: Early Automatic Electrical Cut-Out Patent [34]

One of the earliest US patents to incorporate attributes of a modern day circuit breaker was filed on August 19, 1926 by Heinrich Schachtner of Mannheim, Germany. A portion of the filed schematic can be seen in Figure 2.4, with a full schematic available in Appendix B: Schematics. This design incorporates a push button reset of an automatic electric cutoff as well as thermal cutoff.

In patent US1629640 Schachtner explains the fault conditions and actions, "The cut-out movement of the first slide is commenced when the circuit is heavily overloaded, as by a short circuit, by the excitation of the solenoid coil which is included in the circuit and by the movement of the iron core of the solenoid which is thus produced, and the movement of the slide is suddenly completed by the said mechanical device which is thus set into operation. In addition to the solenoid, a thermostat may be provided consisting of a bi-metal spring and an electric heating element, which latter starts the movement of the slide when the circuit is only slightly overloaded for a fairly long period by reason of the heating action and the curvature of the bi-metal spring thereby produced." [35] Furthermore, Schachtner explains the reset action of the device, "Thus, when the overload in the apparatus ceases, switching-on can be effected by the mere pressure of the finger without, the necessity of unscrewing the apparatus from its socket, but on the other hand, while the overload continues one may attempt, without danger, but without success, to switch-on the apparatus." [35]

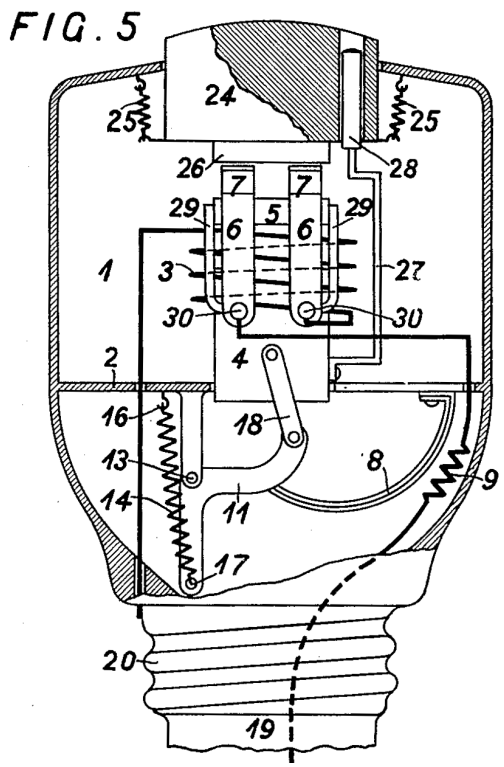


Figure 2.4: Automatic Electric Circuit-Breaker Patent [35]

In general, the electric cutoff was initiated during an overload situation, also known as a abnormally high current situation. The thermal cutoff would only occur after a prolonged high current situation. The prolonged high current would generate heat from the inherent impedance and trip the device. This thermal effect was previously mentioned as "I^2R" losses. These two automatic protection features are incorporated in almost every modern day mechanical circuit breaker.

2.2 Mechanical Circuit Breakers

Mechanical DC circuit breakers, as depicted in Figure 2.5, employ a mechanical switch in the main conduction path and various devices for energy dissipation [36]. The mechanical switch is a very efficient conduction path with contact resistance as low as $10\mu\Omega$, but has a relatively slow opening mechanism [37]. This mechanism is usually spring loaded and the operating times depends on the voltage rating of the system. Typical values in the HVDC range are greater than $10ms$, though some recent mechanical circuit

breaker (MCB) prototypes operate faster [38]. Due to the relatively slow opening of the mechanical switch, an electrical arc is sustained for several moments after operation [39]. This sustained arc is damaging to equipment and unsafe for personnel, making arc suppression very important.

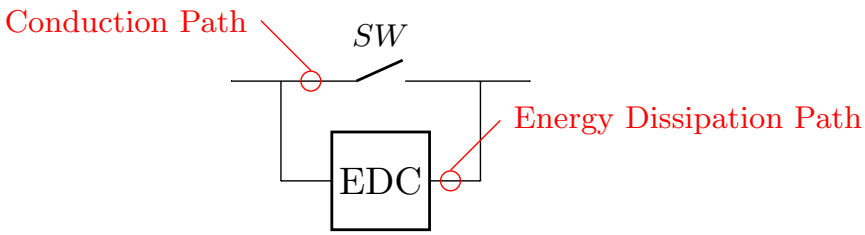


Figure 2.5: Mechanical Circuit Breaker

Arc suppression inside a MCB is commonly completed using only an arc chute, which cools and lengthens the arc [37]. As depicted in Figure 2.6, a cold-cathode arc chute divides and safely extinguishes an electrical arc that is created when the mechanical switch is opened under load [40]. This method is slow, cumbersome, and puts excess stress on the circuit breaker itself.

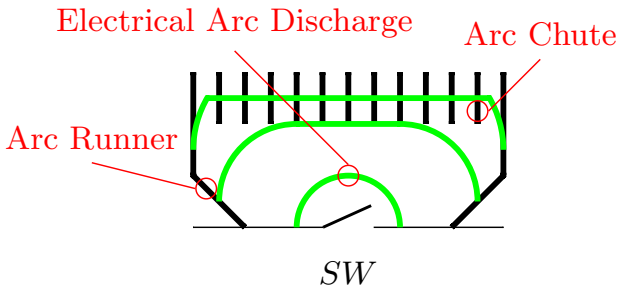


Figure 2.6: Depiction of Arcing in a Cold-Cathode Arc Chute

As discussed in Chapter 1.3, depicted in Figure 1.9, AC has a natural zero crossing which aids in the extinguishing of electrical arcs. In a DC system a natural zero crossing does not exist, but other devices can be put in place to aid in extinguishing electrical arcs or prevent them from forming. This topology of circuit breaker is referred to as a hybrid circuit breaker.

2.3 Hybrid Circuit Breakers

As previously mentioned MCBs do not meet the desired operating parameters for a DC system, but have a very efficient conduction path. Combining dissimilar circuit breaker topologies can lead to a better performing protective device. For instance, installing a MCB in a parallel combination with a resonant current source or solid-state switch. This hybrid circuit breaker can result in a device that outperforms the individual with a highly efficient conduction path and little to no electrical arc during operation.

2.3.1 Mechanical - Resonant Circuit Breakers

One alternative to the MCB is an resonant circuit breaker (RCB) which can employ passive or active resonance components to induce current zero crossings [41]. A simple depiction can be seen in Figure 2.7. The main conduction path is completed with a mechanical switch resulting in high efficiency. The arc suppression is completed by using a resonant current circuit to produce artificial zero crossings by superimposing an oscillating current on the bus current. Resulting in a waveform similar to what is seen in AC voltage systems [42].

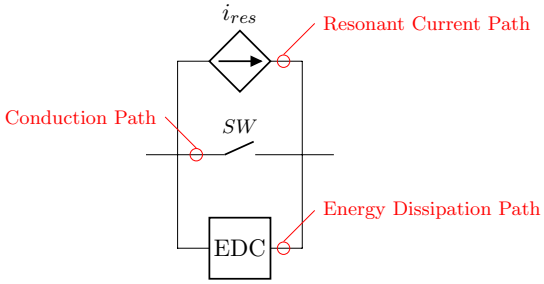


Figure 2.7: Hybrid Resonant Circuit Breaker

This solution quenches the electrical arc after the mechanical switch is opened. The quenching takes place as the electrical arc is forming driving the current through the mechanical switch to zero, allowing the mechanical switch to open under no load. The active resonance arc suppression method is faster than the passive method. This type of arc suppression has been successful in systems operating at $10.5kA$ and completed current

interruptions in under $5ms$ [37]. In short, a RCB overcomes the electrical arcing issues inherent to DC breakers by ensuring little to no electrical arc would occur. Though successful testing has been completed, development of reliable solutions for the current injection circuit have caused difficulty during implementation [31]. A more mature technology currently being used for electric arc discharge mitigation are solid-state devices.

2.3.2 Mechanical - Solid-State Circuit Breakers

Similar to a SSCB, the hybrid circuit breaker consists of series connected solid-state switches in parallel with an energy dissipation component. The defining difference being an additional parallel branch that contains a mechanical switch in series with a single solid-state switch, as seen in Figure 2.8.

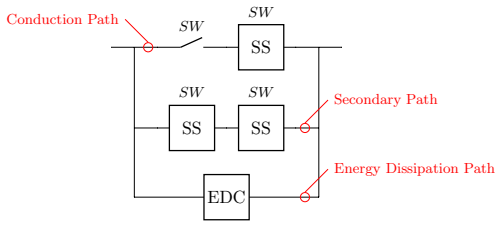


Figure 2.8: Hybrid Solid-State Circuit Breaker

This additional branch containing the mechanical switch creates a circuit breaker type that is a combination or hybrid of MCBs and SSCBs. This hybrid type circuit breaker also combines the key beneficial features from both types as well, being low resistance conduction path and current interruption [36].

The replacement of one solid-state switch with a mechanical switch in the conduction path increased the overall efficiency. This was accomplished by reducing the resistance in the conduction path [43]. The mechanical switch has a very low contact resistance as compared to the relatively high on-state resistance of a solid-state switch. This low conduction resistance is a key feature of MCBs.

The secondary conduction path with only solid-state devices maintains the ability to interrupt current while preventing electrical arc formations. This being a key feature of SSCBs and discussed in more detail in the following section. The normal high on-state

resistance draw back of using solid-state switches is removed, due to the parallel orientation of the this branch to the mechanical switch. Even still this superior configuration of breaker types suffers from high cost and low power density [44].

2.4 Solid-State Circuit Breakers

Solid-state circuit breakers (SSCBs) use solid-state switches, specifically power semi-conductors, to disrupt the conduction path without the generation of an electric arc discharge [44]. A simplified depiction of a SSCB can be seen in Figure 2.9. The main conduction path contains two solid-state switches, with an energy dissipating component in parallel.

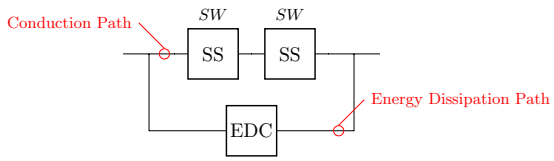


Figure 2.9: Traditional Solid-State Circuit Breaker

The presence of two solid-state switches is necessary to control the bidirectional flow of current. This reasoning will be covered in detail in Chapter 3.2.2 Analysis and Design: Current Flow. The presence of the energy dissipating component is not related to electric arc discharges, but is to prevent any damage from the high voltage pulse produced when direct current through an inductor is interrupted. This concept is also covered in more detail within Chapter 3.4.1 Analysis and Design: Inductive Transient.

As previously mentioned, SSCBs have no moving parts and will not produce electrical arc discharges under normal operating conditions [45]. This ability to interrupt current without using contact separation limits the electrical arc hazard exposure to personnel [46]. As well as increasing the longevity of the equipment by eliminating the arc erosion caused by conductor separation [47]. In turn this characteristic makes the SSCB a prime candidate for a protection device.

In addition to the elimination of electrical arc discharges, the response time of solid-state devices is several orders of magnitude faster than electro-mechanical devices

[48]. This quick response of a few micro-seconds limits the amount of energy in the arc flash hazard exposure as well [45]. Limiting this energy further increases the safety factor of this circuit breaker type, by reducing the peak currents experienced during an interruption transient [39].

The reduction of the peak current experienced during an interruption transient also effects the connecting equipment. This lower peak current could reduce the current withstand and energy absorption requirements for the associated power equipment. As well as reducing the necessary interruption times for downstream loads [49].

An additional benefit SSCBs provide, though minor and very specific to shipboard use, is silent operation [46]. Due to the fact there is no moving parts, SSCBs can be used to realign electrical distribution without fear of producing unwanted sound from mechanical spring discharges or contact closing. This silent operation could give more flexibility to electrical operators during critical operations.

Though there are several advantages to using SSCBs, this circuit breaker type is not without draw backs. In particular, the steady-state efficiency of a SSCB is the most prominent disadvantage for this technology. As seen in Figure 2.9, the solid-state devices are always in the conduction path. Therefore any inherent impedance in the device will reduce efficiency [45]. This concept of efficiency loss was introduced in the previous chapter as line losses or "I^2R" losses. As noted, these conduction losses are significant enough to deteriorate the efficiency of power delivered at medium voltage levels and will only increase at high voltage distribution levels, though research is being done to combat these disadvantages.

Several avenues are being pursued into the reduction of the on-state resistance of solid-state power devices [50]. These same requirements also benefit solid-state switches that are used in high-frequency, high-voltage switching power converters, making them more rugged, efficient, and compact [51]. As well as reducing the on-state resistance, areas of topology design are being pursued in the hopes of mitigating on-state resistance all together. As previously mentioned in hybrid solid-state circuit breakers, one way to reduce the impact of the on-state resistance is to remove the number of devices in the conduction path [43].

2.4.1 State-Of-The-Art Research

The following sections highlight current efforts in the field of SSCB design and component validation. Specific areas of focus are power loss, efficiency, topology, and component choice. Included in the component choice discussion will be the performance of voltage overshoot suppression components.

2.4.1.1 Solid-State Circuit Breaker Protection for DC Shipboard Power Systems: Breaker Design, Protection Scheme, Validation Testing

In this publication, Li Qi and team state, "The ultra fast protection speed requirement motivates the adoption of solid-state circuit breakers (SSCBs) for dc shipboard power systems." [52] Li Qi further asserts that, "semiconductor switching devices have conduction power losses much higher than a traditional electromechanical contact, the choice of the semiconductor technology and components for a solid-state breaker application is of prime importance" [52]. Their solution was based on a reverse blocking integrated gate commutated thyristor (RB-IGCT) in a parallel bidirectional topology, as seen in Figure 2.10, (b) [52].

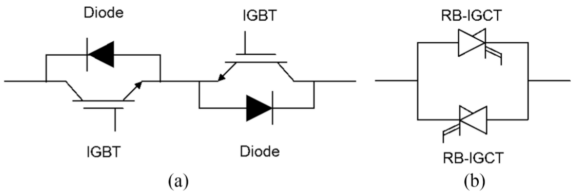


Figure 2.10: Bidirectional switch based on (a) IGBTs and (b) IGCTs. [52]

The conduction power loss experienced by bidirectional switches based on the RB-IGCT, insulated gate bipolar transistor (IGBT), and asymmetrical integrated gate commutated thyristor (A-IGCT) configurations can be seen in Figure 2.11. From this figure, Li Qi concluded the RB-IGCT configuration is optimized throughout the whole current range [52]. Then proceeded to integrate the semiconductor into a DC SSCB topology.

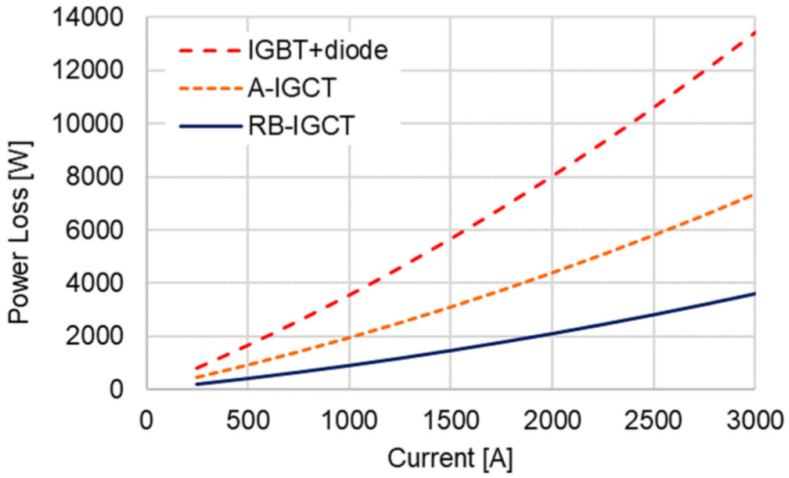


Figure 2.11: Comparison of conduction loss of RB-IGCT against other semiconductors. [52]

The integrated design was a two pole RB-IGCT based SSCB circuit topology, as seen in Figure 2.12. Li Qi states, "In this configuration, only two RB-IGCT in antiparallel configuration are necessary per each pole. Moreover, for each pole, only on[e] RB-IGCT is conducting at the time, which means that only the voltage drop of one device need to be considered in the conduction loss calculation." [52]

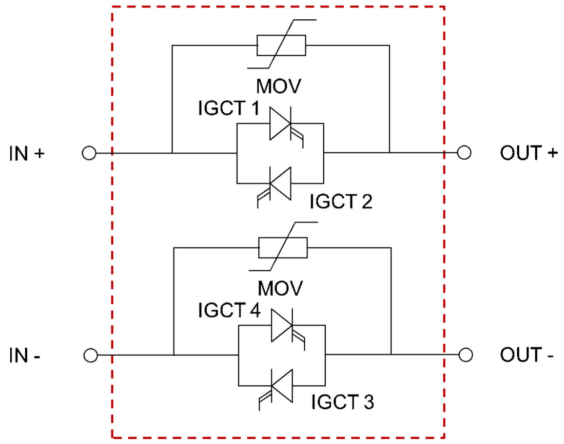


Figure 2.12: Two pole RB-IGCT based SSCB circuit topology. [52]

Furthermore, this topology also employed a metal-oxide varistor (MOV) as a voltage overload protection device. As stated by Li Qi, "Because of the nature of dc systems and the absence of zero current crossing, the inductive energy in the system at the time of opening must be dissipated in order to drive the dc current to zero." [52]

Their choice of a MOV is because it, "is a device that can easily and effectively dissipate relatively large amounts of energy and effectively clamp the voltage to a level that does not damage the semiconductor device that is turning-OFF the current running through the system." [52] This concept of over voltage protection from inductive energy will be explored in Chapter 3.4.4.5: Analysis and Design, Section: 3.4.1.

The efficiency of the integrated design was near 99.9% across the tested current range, as can be seen in Figure 2.13. From these results Li Qi concluded that the RB-IGCT semiconductor was their component choice due to low conduction losses and bidirectional current applications. Furthermore, Li Qi states this device, "guarantees efficiency higher than 99.9% at currents up to 1500 A for a single pole" and "For the SS DCCB with two conducting and breaking poles, the efficiency is higher than 99.8% for currents up to 1500 A" [52].

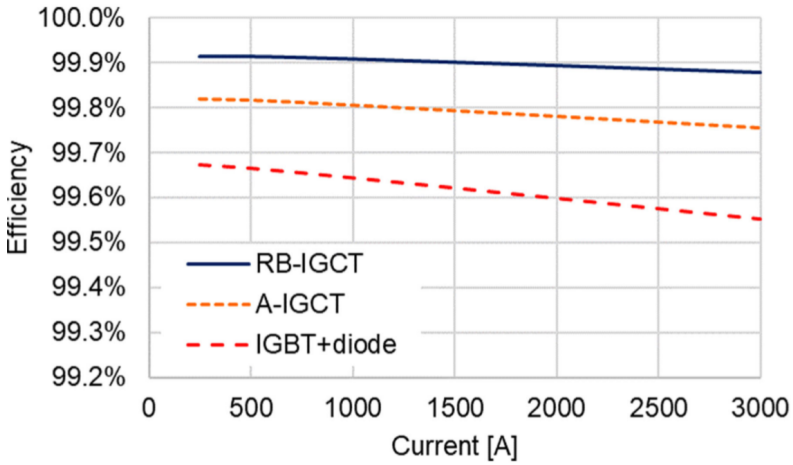


Figure 2.13: Efficiency profile of the RB-IGCT for currents up to 3000 A. [52]

In this publication component conduction losses drove the discussion. This was not only because they are directly linked to efficiency and transmission cost, but could be linked to other design considerations, such as thermal requirements. Similarly Li Qi states, "We can conclude that performance of the RB-IGCT is first in class in terms of efficiency, thus enabling exceptional power density or less intensive cooling and heat handling requirements." [52] This thought process of linking efficiency to thermal requirements will be explored in Chapter 3.4.4.5: Analysis and Design, Section: 3.3.2.

2.4.1.2 A SiC JFET-Based Solid State Circuit Breaker With Digitally Controlled Current-Time Profiles

The research focus of this publication is, "a self-powered solid-state circuit breaker (SSCB) with a digitally controlled current–time profile for both ultrafast short-circuit protection and overcurrent protection" [53]. The development of the current-time profile was completed using the transient thermal properties of silicon-carbide (SiC) junction field-effect transistors (JFETs) [53]. These thermal properties discussed are of interest to this thesis because SiC JFETs are being considered as primary components. Additionally, some primary components will be simulated beyond the manufacturer stated junction temperature limit.

Firstly, an electrical equivalent thermal model was developed considering package forms for a SiC JFET. In this model thermal resistances and thermal capacitances make up each physical layer from the chip to the heatsink. This thermal 'RC' equivalent model can be seen in Figure 2.14.

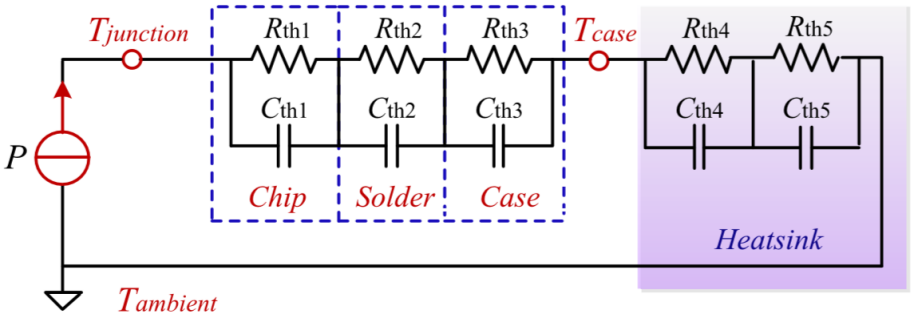


Figure 2.14: Thermal network of the SiC JFET considering package characteristics. [53]

Secondly, using the previous thermal equivalent model a mathematical basis for junction temperature was developed from on-state resistance and power loss. This basis can be seen in Figure 2.15; where on-state resistance is dependent on temperature, power loss is dependent on drain-source current and on-state resistance, and junction temperature is dependent on power loss.

Finally, the change in junction temperature over time can be determined from different drain-source currents. As stated by Dong He, "In order to evaluate the junction

$$R_{ON}(T_j) = R_{ON,300K}\left(\frac{T}{300}\right)^{2.4}$$

$$P(T_j) = I_{DS}^2 R_{ON}(T_j)$$

$$T_j(t) = P(T_j) \times \sum_{i=1}^{3} R_{thi}\left(1 - e^{-\frac{t}{R_{thi}C_{thi}}}\right) + T_{case}$$

Figure 2.15: Mathematical basis for junction temperature. [53]

temperature variation of SiC JFET devices under different overcurrent conditions, the junction temperature curves of SiC JFET devices under 1.5×, 2×, and 3× of the nominal current are obtained by MATLAB" on the basis of the equations displayed in Figure 2.15 and with the results shown in Figure 2.16 [53].

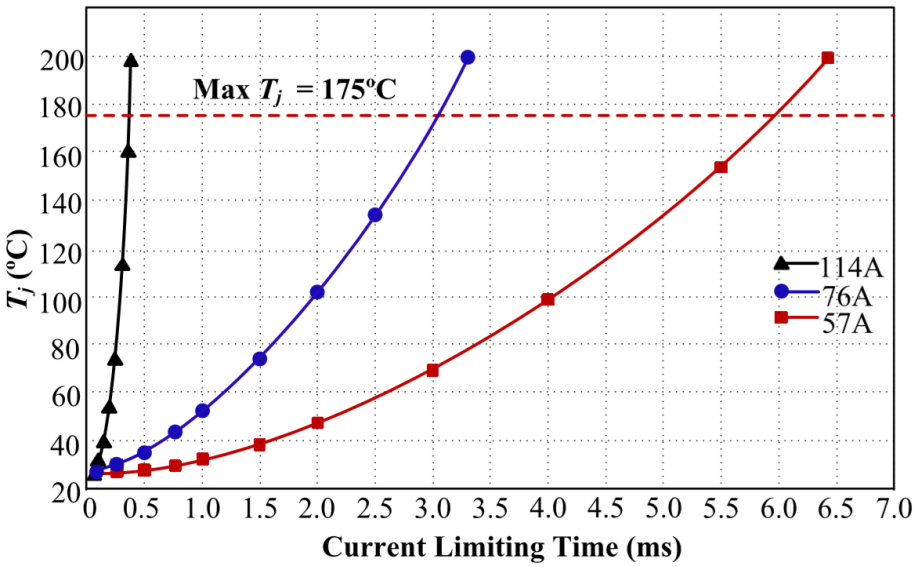


Figure 2.16: Calculated junction temperature with different overcurrents. [53]

In regards to this thesis, the conclusion of higher drain-source currents lead to higher junction temperatures is relevant. It may be necessary to operate primary components past their stated limits. In doing so, it would be pertinent to expect an impact on operating efficiency and this impact should be measured.

2.4.1.3 Voltage Overshoot Suppression for SiC MOSFET-Based DC Solid-State Circuit Breaker

Voltage overshoot during operation can pose a significant problem for SSCBs as discussed earlier. This issue is reiterated by Xinglin Liao, "The transient overvoltage and oscillation phenomena, which are caused by its high switching speed in a solid-state dc circuit breaker based on the silicon carbide (SiC) metal–oxide–semiconductor field-effect transistor (MOSFET), are a crucial problem" [54].

The proposed solution in this publication is a snubber MOV circuit, "based on the voltage ratio of a snubber and an energy-absorbing MOV" [54]. This solution is depicted in a typical topology in Figure 2.17. Which consists of a "dc voltage source generated by ac/dc converters, a SiC MOSFET-based solid-state switch, and an MOV-based absorption circuit" [54].

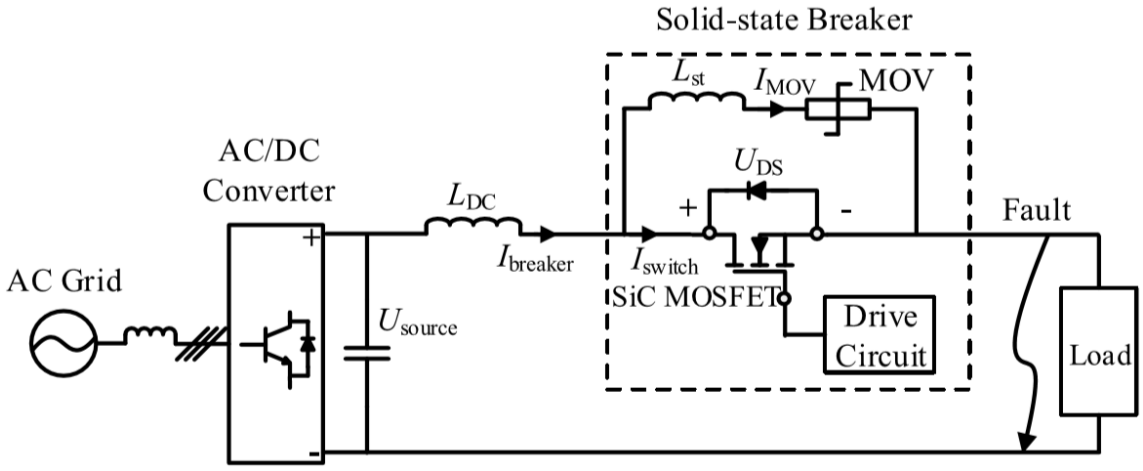


Figure 2.17: Typical topology of a solid-state dc circuit breaker. [54]

A simulation model can be developed from the previously determined typical topology. Using this simulation model and establishing operating parameters, turn-off waveforms can be determined. In this publication a supply voltage of $360V$ and fault current of $11.5A$ are used in conjunction with a K385 MOV [54]. This MOV has a maximum allowable DC voltage of $358V$ [54]. In Figure 2.18 (a), the full sequence interruption of the K385 is shown with Figure 2.18 (b) showing the "zoomed-in illustration of the current commutation" [54].

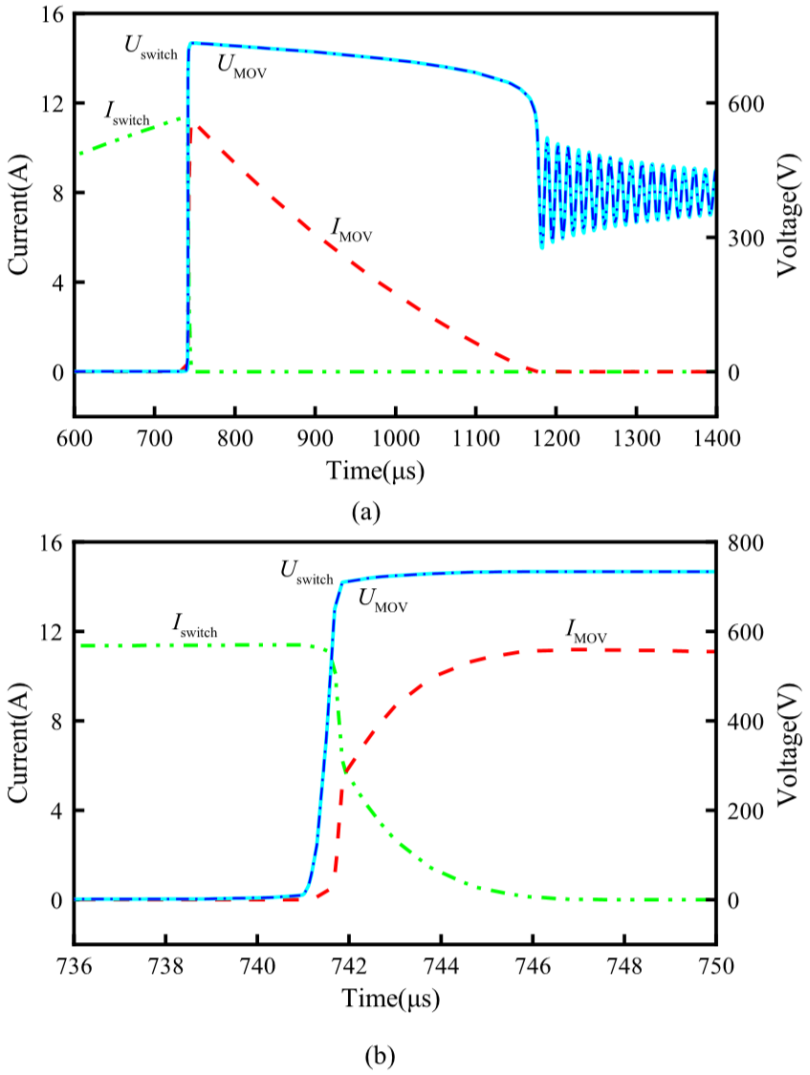


Figure 2.18: Simulation waveforms of the dc circuit breaker without stray inductance. (a) Full sequence interruption. (b) Zoomed-in illustration of the current commutation. [54]

As can be seen in 2.18, the voltage felt by the switch is clamped by the MOV preventing any overvoltage condition. This result is significant to this thesis because an investigation into what components best prevent voltage overshoots will be completed. Therefore, a similar methodology will be conducted as seen in Chapter 3.4.4.5: Analysis and Design, Section 3.4.2.

It is also worth noting, that in this publication simulation solver time step played a significant role. As noted by Xinglin Liao, "In the simulation, the time step has the significant effects on the accuracy and simulation time. If the time step is too large, the

accuracy of the results would be reduced, and the waveforms are not smooth. If the time step is too small, the simulation time is long. Thus, a tradeoff needs to be considered when we set the time step. Through a multitude of attempts, the time step is ultimately set to 50 ns." [54] Similarly, this issue will be expected to be just as significant in the research contained within this thesis.

2.5 Synopsis

Overall, there are multiple ways to disrupt current flow during a transient, even more so when including the many combinations of devices. An expanded landscape of devices, beyond the few covered in this thesis, is shown in Figure 2.19. Though, this depiction is not comprehensive in scope and the field of DC circuit breakers continues to grow.

In review, the mechanical circuit breaker provides a short term low cost solution, but is an ineffective protection device due to the slow response time and large amounts of energy discharged during operation. The hybrid mechanical-resonant circuit breaker is effective at reducing the energy discharged during operation of a mechanical switch and has a relatively fast operating time. Though, the lack of widespread adoption calls in question its use as a protective device at this time.

The traditional solid-state circuit breaker provides very effective protection, due to the fast response time and lack of electric arc discharge. The disadvantages in using a traditional SSCBs is the initial cost of the devices and the relatively low steady-state efficiency. However, the combination of devices as in the case of a hybrid mechanical-SSCB can improve the efficiency without jeopardizing the benefits of using a traditional SSCB.

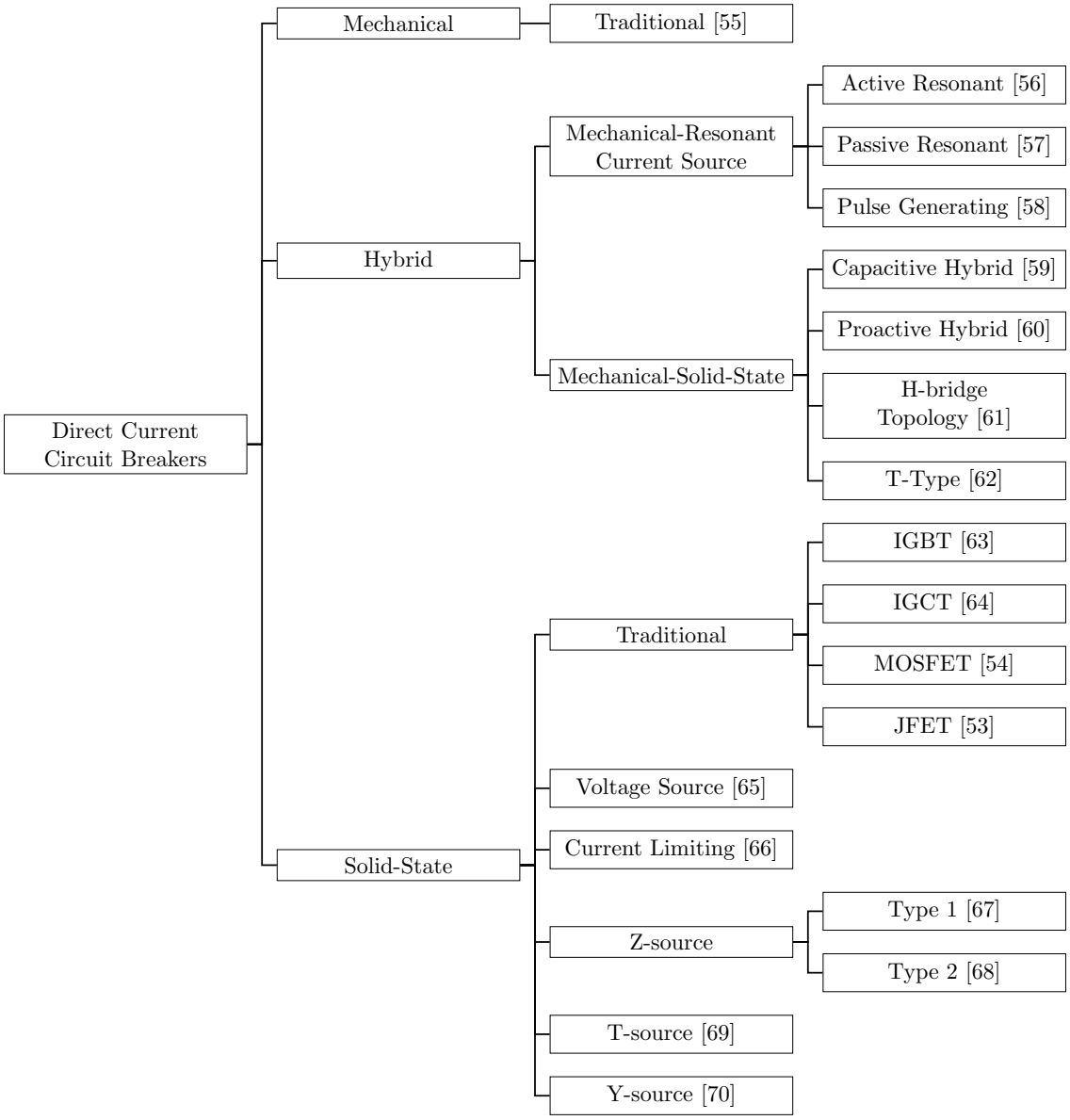


Figure 2.19: Direct Current Circuit Breaker Landscape

CHAPTER 3
ANALYSIS AND DESIGN

The following analysis will focus on the analytical relationships necessary to make informed decisions on the primary component choices of a traditional SSCB. The analytical basis will be fully explained and applied to any circuit configuration deemed necessary for accurate data collection. Any limiting parameters or test conditions will be explained as needed.

3.1 Analysis and Design Format

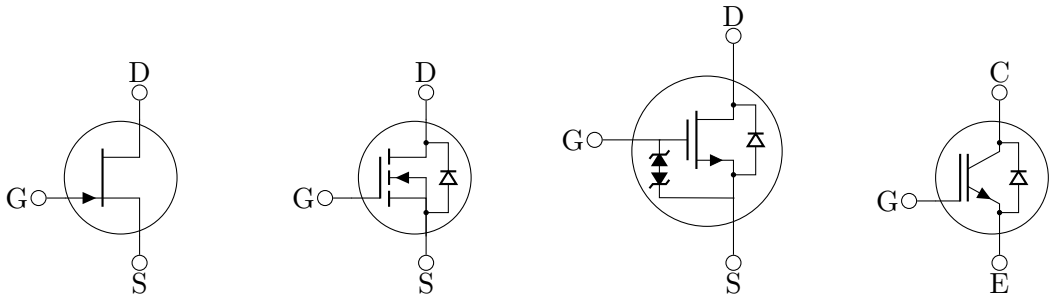
The primary components under investigation are located in the main conduction path and in the energy dissipation path. The main conduction device in a traditional SSCB is a solid-state switch. The energy dissipation circuit (EDC) in a traditional SSCB can be multiple devices or any combination of device types.

Prior to any analysis of solid-state switches, an explanation of the type of devices deemed suitable for use in this thesis will be completed. In this explanation the specific package types of each device will be covered including the associated components. Along with device package types, a discussion of circuit configurations will be completed. This discussion will include proper device orientation for use in a SSCB and appropriate data test point locations. Following, will be the analysis of the solid-state switch devices. This discussion will include how the steady-state efficiency was determined for each circuit configuration previously discussed. In addition, the thermal requirements and cost consideration of each device will covered.

Before any analysis of energy dissipation devices can be conducted, an investigation into the type of devices deemed appropriate for use in this thesis will be completed. This investigation will discuss the operating challenges faced by these devices; in conjunction with, the advantages and disadvantages of each device type. Once an energy dissipation device type has been chosen a methodology for device selection will be prescribed. This methodology will result in a defined selection criteria followed by a comparison analysis for each specific device.

3.2 Solid-State Switch Package Considerations

Any solid-state device installed in the main conduction path of a SSCB would require the ability to pass a significant amount of current, have a relatively low on-state resistance, and have a large voltage withstand limit. There are at least four types of solid-state devices that would meet the demands of a traditional SSCB. These include the JFET, metal-oxide semiconductor field-effect transistor (MOSFET), a co-packaged cascode configuration of a JFET and MOSFET, and an IGBT [71]. A simplified representation of each of these device types can be seen in Fig. 3.1a, Fig. 3.1b, Fig. 3.1c, and Fig. 3.1d respectively.



(a) JFET [72] (b) MOSFET [73] (c) Cascode Device [74] (d) IGBT [75]

Figure 3.1: Schematic Symbols of Device Types

Each of these device types have their own attributes that make them uniquely suited for use inside a SSCB. The SiC variety of JFETs normally exhibit very low on-state resistance, ($R_{DS(ON)}$), with the capability of conducting a significant amount of current, (I_D). This combination will prove useful in achieving high efficiency and will be covered in more detail in later sections. However, SiC JFETs differ from MOSFETs in that there is no body-diode and normally require a freewheeling-diode to be installed anti-parallel to the device [76]. It is also worth noting that a SiC JFET packaged alone is a normally-on device and would not provide satisfactory circuit protection installed as-is into a SSCB.

A normally-off configuration is important for use inside a protective device. If control power is removed from the device, the SSCB would open. This ability to fail open would provide necessary protection for any equipment or personnel down stream of the

device. Conversely, a normally-on device would close on loss of control power. This would allow current to pass to any equipment or personnel downstream of the device. This could cause inadvertent power transients resulting in equipment damage or personnel injury.

Another normally-off device type to be considered is a power MOSFET. These devices typically have a high withstand or blocking voltage, (V_{DS}), coupled with the capability of conducting a significant amount of current, (I_D). However as previously noted MOSFETs inherently contain a parasitic device known as a body-diode. This parasitic device is a result of the manufacturing process of power MOSFETs and is not a separate device, but considered to be a permanently installed anti-parallel diode [76].

An additional device type to be considered would be a co-packaged cascode configuration of a JFET and a MOSFET. This SiC field-effect transistor (FET) device is a unique circuit configuration in which a normally-on SiC JFET is co-located with a low-voltage MOSFET in a single device package. This combination of devices provides the same abilities as the previously mentioned normally-on SiC JFET, but in a normally-off configuration with the addition of an inherent anti-parallel diode.

A final device type being investigated is an IGBT. These devices have a similar control scheme to MOSFETs, but in general have a higher voltage, (V_{CE}), and current, (I_C), capability. Similarly to the previously discussed device types though not inherent, IGBTs are normally packaged with an anti-parallel diode. Conversely, IGBTs do not have a static on-state resistance. An equivalent on-state resistance for each specific device can be estimated using the manufacturer provided typical output characteristic curve. Also known as the "I-V" curve or more specifically the collector current versus the collector-emitter voltage curve.

3.2.1 Anti-Parallel Diodes

Most modern manufactured solid-state switch package designs have the option to include diodes installed anti-parallel to the solid-state switch. This package option is commonly referred to as a module and the FET device alone is usually referred to as a discreet device. The diodes installed into modules are referred to as freewheeling diodes, but have many names including flyback diodes, snubber diodes, suppressor diodes, catch

diodes, clamp diodes, or even known as commutating diodes. The purpose of a freewheeling diode is to allow current to flow in reverse bypassing the solid-state switch; for any FET it would be from the source pin to the drain pin, as seen Figure 3.2.

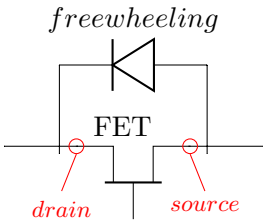


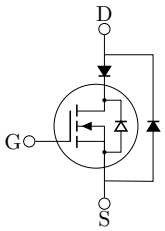
Figure 3.2: Field-Effect Transistor Representation

The body-diode present in a power MOSFET is similar to a freewheeling diode in action, but is not purposely installed. Rather it is a result of the device material orientation. Inside a MOSFET there exists a p–n^- junction, which makes up the body-diode. This junction has a parasitic attribute that forms an effective diode anti-parallel with the MOSFET.

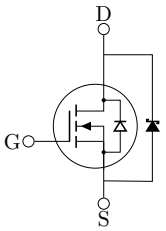
In silicon (Si) MOSFETs the body-diode is capable of conducting the same amount of current as the MOSFET but the switching speed of the body-diode is slower resulting in high peak currents during diode turnoff [76]. These currents can be higher than the rated current capability of the MOSFET and result in device failure. Due to the likely hood of device failure most Si MOSFETs are offered in a module package with a fast recovery diode installed parallel to the body-diode with a blocking diode preventing the body-diode from turning on, as seen in Fig. 3.3a.

Conversely, SiC MOSFETs have low on-state resistance and inherently low switching loss due to the wider band gap of SiC [77]. These SiC power MOSFETs are offered in a module package with only a fast recovery diode installed parallel to the body-diode, as seen in Fig. 3.3b. Otherwise they are noted as discreet devices and have the same circuit symbol as Fig. 3.1b.

The performance of anti-parallel diodes is crucially important when using solid-state devices for voltage regulation. For the purpose of a SSCB anti-parallel diodes are not required and add nothing to the functionality of a SSCB. Though their mere



(a) Silicon MOSFET with Fast Recovery Diode Circuit [76]



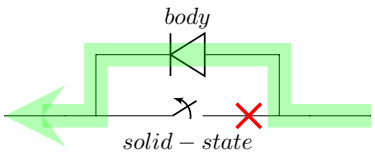
(b) SiC MOSFET with Fast Recovery Diode Circuit

Figure 3.3: MOSFETs with Fast Recovery Diode Circuit

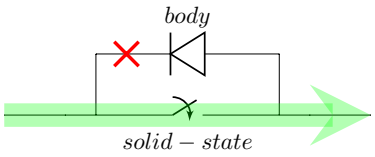
presence must be taken into account when analyzing the performance of a SSCB or when designing the operating limits of an EDC. In particular, if the reverse break-down voltage of the diode is less than the activation voltage of the EDC. In that case, the EDC would never turn on forcing the MOSFET to handle any transient alone.

3.2.2 Current Flow

Each of the solid-state switches under consideration can block voltage and current in only one direction. They are all considered to be unidirectional devices. For example, the power MOSFETs can prevent the current from the supply side being passed to the load side, but the load side could still pass current through the body-diode onto the supply side. Therefore current is only blocked one way, hence unidirectional. This can be seen in Figure 3.4. This device orientation is also considered to be a two-quadrant switch [76].



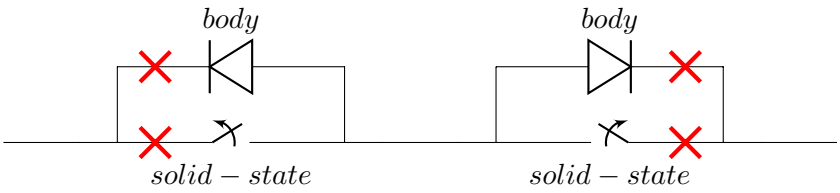
(a) Open Switch Current Flow Depiction



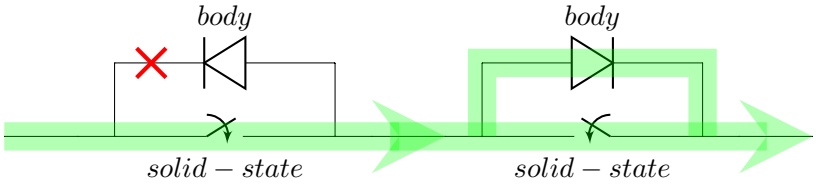
(b) Closed Switch Current Flow Depiction

Figure 3.4: Simplified Unidirectional Solid-State Switch Current Flow

If two unidirectional devices are placed in series in reverse conduction orientation, it is considered a bidirectional switch formation. Using the same example as before, the first power MOSFET would block any current from the supply side and the second power MOSFET would block any current from the load side. Therefore current is blocked two ways, hence bidirectional. This can be seen in Figure 3.5a. This device orientation is also considered to be a four-quadrant switch [76].



(a) Open Switch Current Flow Depiction



(b) Closed Switch Current Flow Depiction

Figure 3.5: Simplified Bidirectional Solid-State Switch Current Flow

As previously stated the unidirectional circuit can not fully disconnect the load from the source. Therefore to ensure a proper disconnection between the source and load a bidirectional circuit must be used. This in turn means there will be three components conducting current and affecting efficiency, as can be seen in Figure 3.5b. When the load side switch is closed and conducting current any body-diode present would be in parallel to the closed switch and would pass some current, albeit small. Taking into account the small amount of power being consumed by the body-diode as well as both closed switches is important to accurately calculate the efficiency of the bidirectional circuit. This also means that simulating a bidirectional circuit alone would not produce an appropriate representation of individual switch efficiency. Therefore, both unidirectional and bidirectional circuits will be simulated to produce more accurate data points for component choice.

3.2.3 Selection Process

The following process depicted in Figure 3.6 will be used to down select between solid-state devices of the previously defined types. As a reminder, these include JFETs, MOSFETs, co-packaged cascode configuration devices, and IGBTs. The selection pool will consist of devices that are readily available from electronic component distributors.

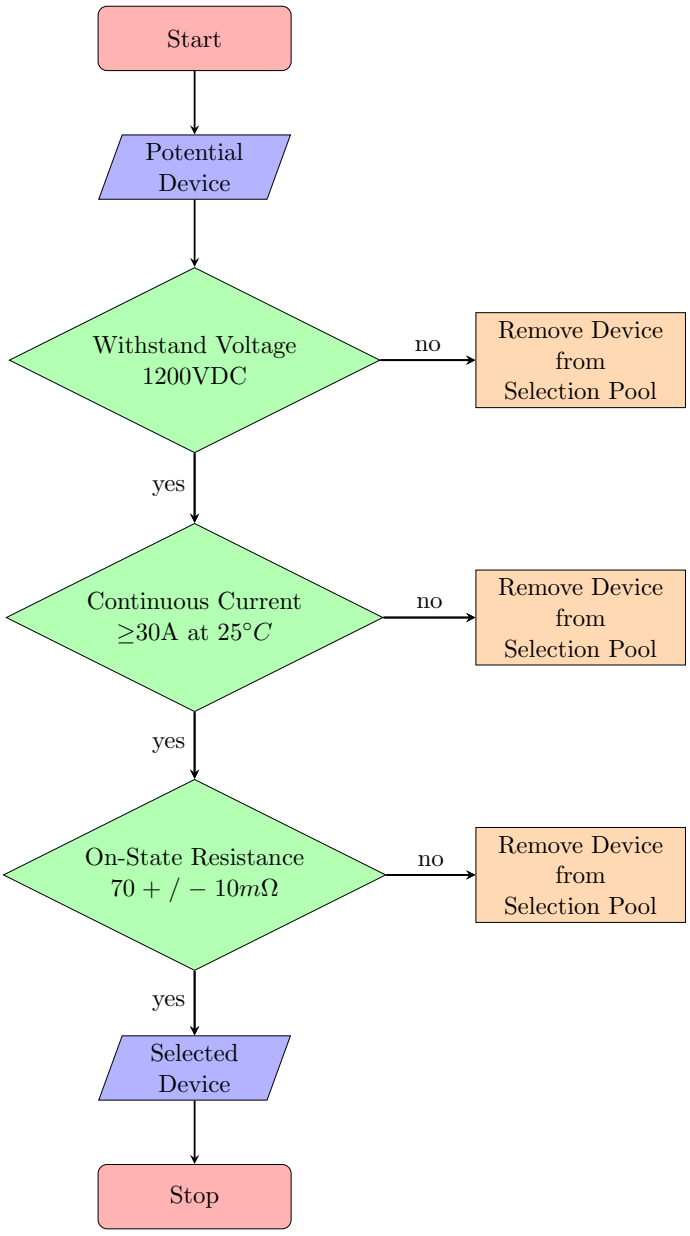


Figure 3.6: Solid-State Switch Device Selection Flowchart

3.3 Solid-State Switch Circuit Analysis

A diagram of the unidirectional circuit that will be used to determine the efficiency of each solid-state switch can be seen in Figure 3.7. Here the solid-state switch is depicted as a MOSFET and will pass current from a DC voltage source to a resistive load. Additionally, a gate circuit will be required to control the potential applied to the gate pin of each solid-state switch. This gate circuit will consist of an independent DC voltage source and in most cases will require an external gate resistance. The gate circuit component values will be determined by the manufacturer data sheet or otherwise modified as necessary for proper operation in this application. If the manufacturer's data sheet recommends a gate resistance of 0Ω, then the external resistance will be completely removed.

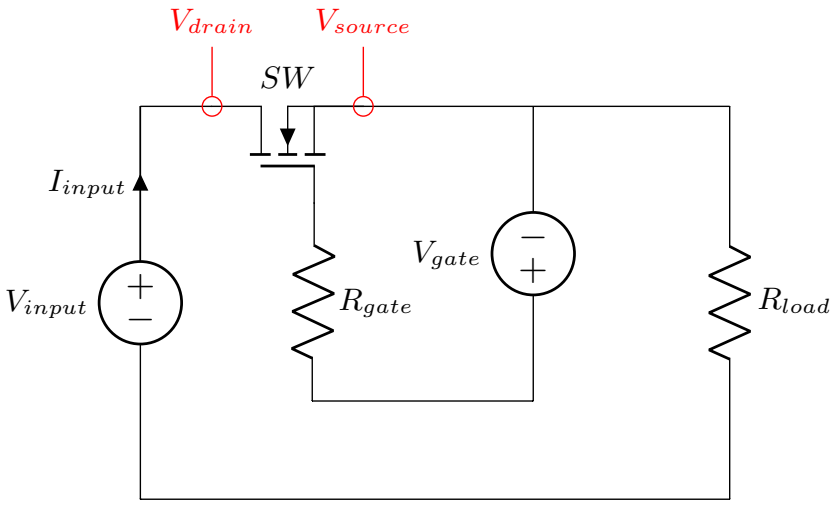


Figure 3.7: Unidirectional Solid-State Switch Circuit

A diagram of the bidirectional circuit that will be used to determine the efficiency of each solid-state switch pair can be seen in Figure 3.8. Similarly to the unidirectional circuit, the solid-state switch pairs are depicted as MOSFETs and will pass current form a DC voltage source to a resistive load. Just as before, a gate circuit will be required for each solid-state switch. It is possible to operate the solid-state switch pair with a single gate circuit. This could be investigated as future work, though in this case the doubling of the necessary external resistances and independent voltage sources will be required.

The data test point locations for each circuit configuration are labeled in each figure. These test point locations will correspond to the following efficiency analysis below in sub-section, 3.3.1 Efficiency. It is worth noting that in the bidirectional circuit configuration a single test point is used between the source pins of the installed solid-state device pair.

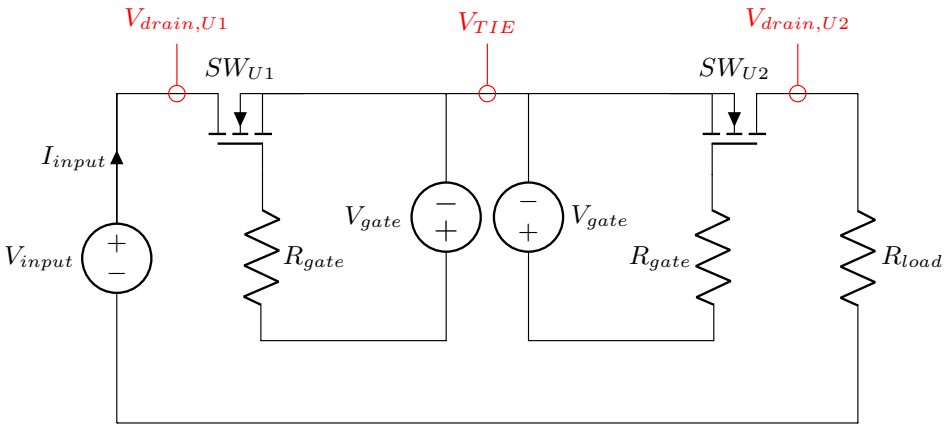


Figure 3.8: Bidirectional Solid-State Switch Circuit

3.3.1 Efficiency

The efficiency of a component is determined by the ratio of input power, (P_{input}, $watt$), to output power, (P_{output}, $watt$). A component with no loss would have a ratio of one, and therefore an efficiency of one hundred percent. This relationship can seen in Equation 3.1 [78]. Referencing back to the Introduction, Chapter 1.3, the highest efficiency as possible is desirable. In other words, the higher the efficiency the lower the losses. Therefore the lower required total energy needed to be produced to meet the energy demand and overcome the energy loss.

$$efficiency = \frac{P_{output}}{P_{input}} \times 100\ [\%] \tag{3.1}$$

The efficiency percentage can be further described by defining the input power, (P_{input}, $watt$), to the component as seen in Equation 3.2. This equation is derived from the

power law as mentioned in the Introduction, Chapter 1.3, Equation 1.3. The input power as defined in Equation 3.2 with the input voltage, (V_{input}, $volt$), and input current, (I_{input}, $ampere$), is applicable for the unidirectional circuit, as well as the bidirectional circuit.

$$P_{input} = V_{input} \times I_{input}\,[W] \tag{3.2}$$

Similarly, the output power definition of each circuit configuration is the same, though the definition of each power loss is unique. In the case of the unidirectional circuit, Figure 3.7, the output power, (P_{ouput}, $watt$), is determined by subtracting the power consumed by the solid-state switch, (P_{loss}, $watt$), as seen in Equation 3.3. The power loss, (P_{loss}, $watt$), is determined by using the input current, (I_{input}, $ampere$), and the voltage potential difference from the drain pin of the solid-state switch (V_{drain}, $volt$) to the source pin of the solid-state switch, (V_{source}, $volt$).

$$\begin{aligned} P_{output} &= P_{input} - P_{loss}\,[W] \\ P_{loss} &= (V_{drain} - V_{source}) \times I_{input}\,[W] \end{aligned} \tag{3.3}$$

Another method that can be used to determine the power dissipated by the switch would be to use it's Ohmic losses or the "I^2R" value. This value can be calculated using the manufacturer listed on-state resistance value, ($R_{DS(ON)}$), multiplied by the squared drain current, (I_D) [78], as seen in Equation 3.4.

$$P_{loss} = R_{DS(ON)} \times (I_{input})^2\,[W] \tag{3.4}$$

This method however relies on a static data point in addition to the reporting accuracy of the manufacturer's data sheet to match the provided Simulation Program with Integrated Circuit Emphasis (SPICE) model. Any dynamic characteristics in the model as well as errors, estimations, or rounding by the manufacturer would translate into the calculations. Calculating the power dissipated by measuring the voltage and current within the simulation reduces the potential for introduced inaccuracies.

In regards to the bidirectional circuit, Figure 3.8, the output power, (P_{ouput}, $watt$), is determined by subtracting the power consumed by each solid-state switch, ($P_{loss,U1}$, $watt$) and ($P_{loss,U2}$, $watt$), as seen in Equation 3.5. The power loss of the first solid-state switch, ($P_{loss,U1}$, $watt$), is determined by using the input current, (I_{input}, $ampere$), and the voltage potential difference from the drain pin of the first solid-state switch ($V_{drain,U1}$, $volt$) to the switch tie point, (V_{TIE}, $volt$). The power loss of the second solid-state switch, ($P_{loss,U2}$, $watt$), is determined by using the input current, (I_{input}, $ampere$), and the voltage potential difference from the switch tie point, (V_{TIE}, $volt$) to the drain pin of the second solid-state switch ($V_{drain,U2}$, $volt$), as determined necessary by expected steady-state current flow direction.

$$\begin{aligned} P_{output} &= P_{input} - P_{loss,U1} - P_{loss,U2}\,[W] \\ P_{loss,U1} &= (V_{drain,U1} - V_{TIE}) \times I_{input}\,[W] \\ P_{loss,U2} &= (V_{TIE} - V_{drain,U2}) \times I_{input}\,[W] \end{aligned} \tag{3.5}$$

The aforementioned efficiency analysis will be applied in the simulation phase. The details of the application will be explained in the following Chapter 4.5.3 Simulation Setup and Test.

3.3.2 Thermal

The efficiency of a component can be directly tied to the waste heat produced by the component. Similar to the line losses mentioned in Chapter 1.3 Introduction, being as current flows through a component and encounters resistance, heat is generated. This heat if not controlled can lead to more impedance and in turn lead to even more heat being produced resulting in a heating runaway effect. This effect reduces the efficiency of the component in the least or at most causes a catastrophic failure of the component.

Controlling the heat generated by components can be accomplished by heat transfer; which is the process of removing the heat produced from a device and adding it to a cooling medium. This relationship in it's simplest form can be seen in Equation 3.6 [79]. Where the heat supplied to a material (Q, J) is equal to it's mass (m, kg) multiplied

by the specific heat capacity of the material in which heat is being transferred through (c, $\frac{J}{kg\,K}$) and the change in temperature (ΔT, $Kelvin$) across that material [80] [81].

$$Q = m\,c\,\Delta T\,[J] \tag{3.6}$$

The process of heat transfer has three main mechanisms: conduction, convection, and radiation. These mechanisms occurring in a component with a heat sink cooling apparatus attached are depicted in Figure 3.9. Here the heat producing component is mounted to a printed circuit board (PCB) with a heat sink mounted to the component. There is an intermediate layer consisting of a thermal interface material (TIM) that facilitates the conduction process from the component to the heat sink.

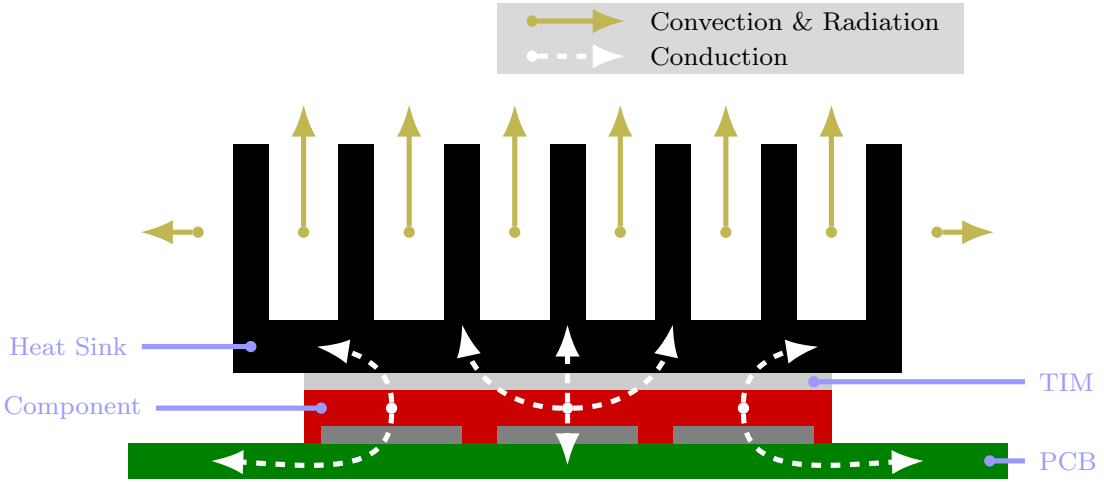


Figure 3.9: Heat Transfer Mechanisms

Conduction is the transfer of kinetic energy between particles. More specifically, "Conduction is the transfer of energy from more energetic particles of a substance to the adjacent less energetic ones as a result of interactions between the particles" [80]. This process does take place in fluids, but it is the main mechanism in solids. The energy transfer within solids is due to the vibrations of molecules within a lattice structure. The process of heat transfer via conduction is expressed using Equation 3.7 [79]. In that the heat transfer rate ($\dot{Q}_{cond}$, W) is the product of the thermal conductivity (k, $\frac{W}{m\,K}$), from a surface area (A_s, m^2), with temperature difference across the material ($T_{Hot} - T_{Cold}$,

$Kelvin$), divided by the thickness of the material (d, m) [80] [81]. This equation can be reduced to the differential form, which is Fourier's law of heat conduction.

$$\begin{aligned} \dot{Q}_{cond} &= k\, A_s\, \frac{T_{Hot} - T_{Cold}}{d}\, [W] \\ \dot{Q}_{cond} &= -k\, A_s\, \frac{dT}{dx}\, [W] \end{aligned} \tag{3.7}$$

Convection is the transfer of energy into a fluid that is in motion when a temperature gradient is present. More specifically, "Convection is the mode of energy transfer between a solid surface and the adjacent liquid or gas that is in motion, and it involves the combined effects of conduction and fluid motion"[80]. Heat transfer via convection can be expressed by Newton's law of cooling as Equation 3.8 [79]. Where the heat transfer rate of convection ($\dot{Q}_{conv}$, W) is equal to the product of the convection heat transfer coefficient (h, $\frac{W}{m^2K}$), surface area (A_s, m^2), and temperature difference between the surface and fluid [80].

$$\dot{Q}_{conv} = h\, A_s\, (T_s - T_f)\, [W] \tag{3.8}$$

Finally, radiation is emitted energy in the form of electromagnetic waves. Radiation can also be defined as "the energy emitted by matter in the form of electromagnetic waves (or photons) as a result of the changes in the electronic configurations of the atoms or molecules" [80]. The process of heat transfer by way of radiation can be represented by Equation 3.9 [79]. Here the rate of heat transferred ($\dot{Q}_{rad}$, W) is the product of the surface emissivity (ϵ), Stefan Boltzmann constant (σ, $\frac{W}{m^2K^4}$), surface area (A_s, m^2), and the absolute temperature difference between the emitting surface and surrounding surface ($T_s^4 - T_{surr}^4$, K^4) [80] [81].

$$\dot{Q}_{rad} = \epsilon\, \sigma\, A_s\, (T_s^4 - T_{surr}^4)\, [W] \tag{3.9}$$

The three main mechanisms of heat transfer are present in all methods of controlling component temperature, but some mechanisms are more dominate than others. In most methods of temperature control, convection is the primary process and

radiation is the secondary process. Conduction is present but normally just facilitates the temperature control process by furring the heat away from the heat producing component to later be removed from the system by way of convection or radiation.

As depicted in Figure 3.9 a component has a heat sink mounted to its case, which in turn is immersed into a moving fluid. The movement of the fluid can be inherit to the heat transfer process by way of natural convection, "alternatively called free convection, where any fluid motion occurs by natural means such as buoyancy" [80]. Where in the fluid surrounding the heat sink will heat up and rise and once cool again will fall returning to the surface of the heat sink to continue the cycle. The movement of the fluid can also be forced, via a pump if the fluid is a liquid or a fan if the fluid is a gas.

The most prevalent method of temperature control among solid-state component designs is forced air convection and therein will be used as an example in the following thermal analysis. Again, as depicted in Figure 3.9 the heat produced by the device is transferred to the case of the device by design, then transferred to a heat sink which is immersed in a cooling medium. A heat sink is a passive component made of material that has a high thermal conductivity coefficient. This material tends to be aluminum or copper, depending on the application and desired cost of the material. The cooling medium in which the heat sink is immersed is a fluid, be it gas or liquid, and generally is air or water. Other fluids can be used and are dependent on the application and cost, including pure hydrogen which is used in industrial electrical motors.

The cooling medium can be left to naturally convect as previously mentioned, or can be forced across the surface of the heat sink. In the most common case of using air as a cooling medium a fan can be used to move the fluid across the surface of the heat sink. This is important to note because the flow rate of the fluid impacts the heat transfer rate positively. Meaning, the higher the fluid flow rate the higher the heat transfer rate. In some cases the fluid flow rate can overcome an ill suited or poorly designed heat sink, but this is not a reliant design method.

Designing an optimum heat sink for each device used is a science in of its self and is beyond the scope of this thesis. However, a delineation of the devices can be made using the thermal properties of each device. The key variable for the purpose of this thesis will

be the thermal resistance from the heat sink to the ambient temperature, ($R_{\theta,sa}$,$^{\circ}C/W$). This attribute will be the driving force for the design of the heat sink and in turn the associated cost.

The derivation of each device's thermal resistance from the heat sink to ambient temperature starts with the identifying the various temperatures at play; including the junction temperature (T_j), case temperature (T_c), heat sink temperature (T_s), and ambient temperature (T_a). The location of these temperatures in a typical system can be seen in Figure 3.10.

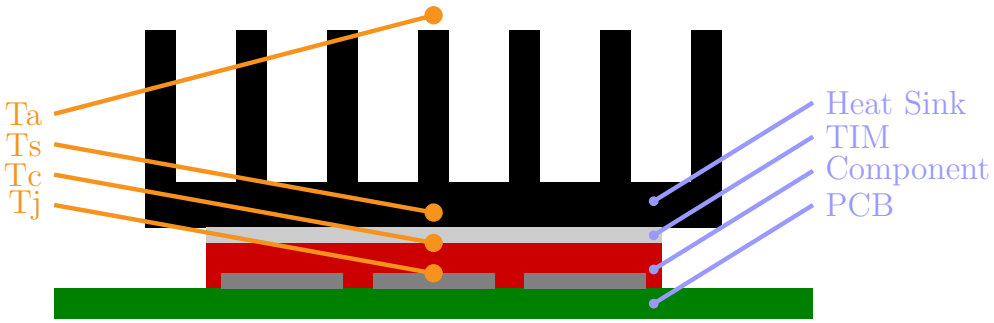


Figure 3.10: Heat Transfer Temperature Locations

Controlling the junction temperature, or channel temperature, is key to ensuring long service life, high efficiencies, and preventing catastrophic failure of the component. The junction temperature (T_j), which will be the highest temperature point inside of a solid-state switch, can be related directly to the power dissipation of the switch (P_{diss}), the thermal resistances of the cooling system (R_θ), and ambient temperature in which the solid-state switch is operating (T_a). This relationship can be seen in Equation 3.10 [82]. The relationship of the thermal characteristics from Equation 3.10 and the temperatures noted in Figure 3.10 can be seen electrically represented in Figure 3.11.

$$T_j = T_a + P_{diss}(R_{\theta,jc} + R_{\theta,cs} + R_{\theta,sa}) \; [^{\circ}C] \tag{3.10}$$

The average power loss or power dissipated of the device, (P_{diss}, $watt$), can be determined similarly to the efficiency of the device. This can be seen in Equation 3.11 with the "I^2R" losses of the device calculated from the on-resistance, ($R_{DS(ON)}$, Ω) and

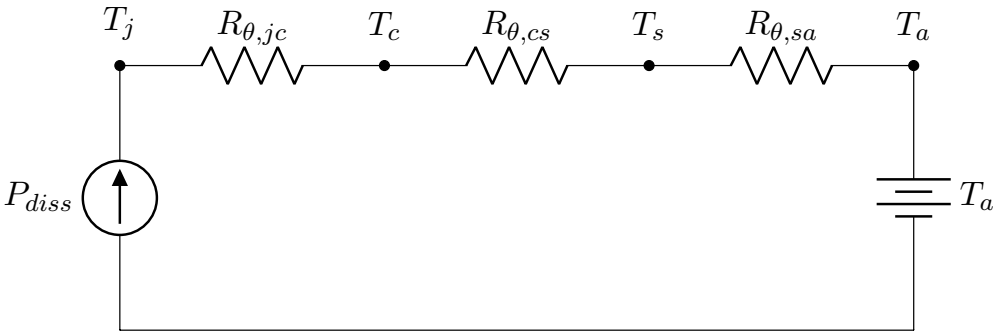


Figure 3.11: Representative Electrical Analog of Heat Transfer [82]

the drain current, (I_D, *ampere*) [78]. The thermal resistance from the junction to the case, ($R_{\theta,jc}$, $^\circ C/W$), can be surveyed from the manufacturer data sheet of each device to be simulated or tested.

$$P_{diss} = R_{DS(ON)} \times (I_D)^2 \, [W] \tag{3.11}$$

Similarly with the thermal resistance from the case to the heat sink, ($R_{\theta,cs}$,$^\circ C/W$); the material used to bond the heat sink to the device will normally list a thermal impedance or thermal conductance on the manufacturer's data sheet. This thermal impedance can be used to determined the thermal resistance by using Equation 3.12 in conjunction with the known surface area of the device package. Therefore the appropriate thermal resistance per each device package type, be it TO-247, SOT-227, or other such package size can be determined as needed.

$$R_{\theta,cs} = Area \times Thermal\ Impedance \, [^\circ C/W] \tag{3.12}$$

To accurately portray the thermal resistance of each solid-state switch the surface area of each package type in contact with the heat sink will be determined from the manufacturer data sheet. These values tend to have a range or a plus/minus for manufacturing tolerances. The average or center value of each reported dimension will be used for each package type. Moreover, the reported units of the thermal impedance is not uniform across all manufacturers. Any reported values that deviate from $^\circ C/W$ will be

converted accordingly. Any manufacturer reported values that use imperial area units will be converted using Equation 3.13. As well as any manufacturer reported values in Kelvin will be converted using Equation 3.14.

$$1\ in^2 = 6.4516\ cm^2 \tag{3.13}$$

$$1\ \frac{K\ cm^2}{W} = 1\ \frac{^\circ C\ cm^2}{W} \tag{3.14}$$

As previously mentioned, for the purposes of this thesis the derived thermal resistance value of each device will be used to compare the thermal design effectiveness of each solid-state device. The thermal resistance from the heat sink to the ambient temperature, ($R_{\theta,sa}$,$^\circ C/W$), is commonly used to determine the design of the heat sink required for the given application. This value can be calculated for a known ambient temperature, (T_a,$^\circ C$) and desired junction temperature, (T_j,$^\circ C$). This is done by rearranging Equation 3.10 to solve for this thermal resistance.

The resultant thermal resistance is a maximum allowable value. Therefore, the larger the value the more flexibility in heat sink design. In turn the smaller the thermal resistance value the more restrictive the cooling options. This may lead to the necessity of a more elaborate and expensive cooling apparatus. A low thermal resistance value may also give insight into the lifetime of the device, be it shortened from thermal stress. The aforementioned thermal analysis will be completed in the simulation phase. The details of the application will be explained in the following chapter, Simulation Setup and Test, Chapter 4.5.3.

3.3.3 Costing

The cost of a component can be described in many ways including a straight forward approach of per unit cost. Using the per unit cost in conjunction with other attributes of the component may provide a more holistic view of the cost of a component. For instance, a particular component may have a lower price per unit, but the current carry capability of the component is much more limited. This may lead to requiring more

devices orientated in parallel in order to lower the current in each conduction branch. This in turn would cause the overall price of the application to increase.

One method to determine if more than one conduction path is required, is to divide the desired full load current, ($I_{desired}$, $ampere$), by the rated drain current, (I_{drain}, $ampere$), of the device, as seen in Equation 3.15. The number of conduction paths required, ($n_{conduction}$), is equal to the result rounded up to the next integer.

$$n_{conduction} = \lceil I_{desired} \,/\, I_D \rceil \tag{3.15}$$

The application of Equation 3.15 requires knowing the desired full load conduction current of the application ahead of time. The desired current would be application specific and therefore change from project to project. Determining the final cost would require translating the number of required conduction paths to price by multiplying by the unit price. This would result in the total cost of solid-state switches for that particular project, as seen in Equation 3.16.

$$Price_{total} = n_{conduction} \times Price_{unit} \,[\$] \tag{3.16}$$

A more universal method is to determine the price per current, ($Price_{ampere}$, $\$/ampere$). This method does not require any foreknowledge of any project and only relies on the attributes of the components themselves. Determining the unit price of a solid-state switch per it's drain current carrying capacity can be completed using Equation 3.17. The unit price, ($Price_{unit}$, $\$/unit$), information can be gathered from an electronic component distributor. The typical drain current limit, (I_D, $ampere$), information of each solid-state switch to be tested can be gathered from the manufacturer provided data sheets.

$$Price_{ampere} = \frac{Price_{unit}}{I_D} \,[\$/A] \tag{3.17}$$

Another aspect of cost when using a device is the associated material and equipment necessary to operate the device within the intended or desired specifications. Specifically, to use a solid-state switch a cooling apparatus is necessary. As previously

described in the Thermal Subsection 3.3.2, a heat sink will be used to ensure proper device junction temperature while in operation. The cost of a heat sink can be correlated to the thermal resistance as seen in Figure 3.12; note this figure was originally produced 1995 and costs have changed over time, though the trend remains unchanged.

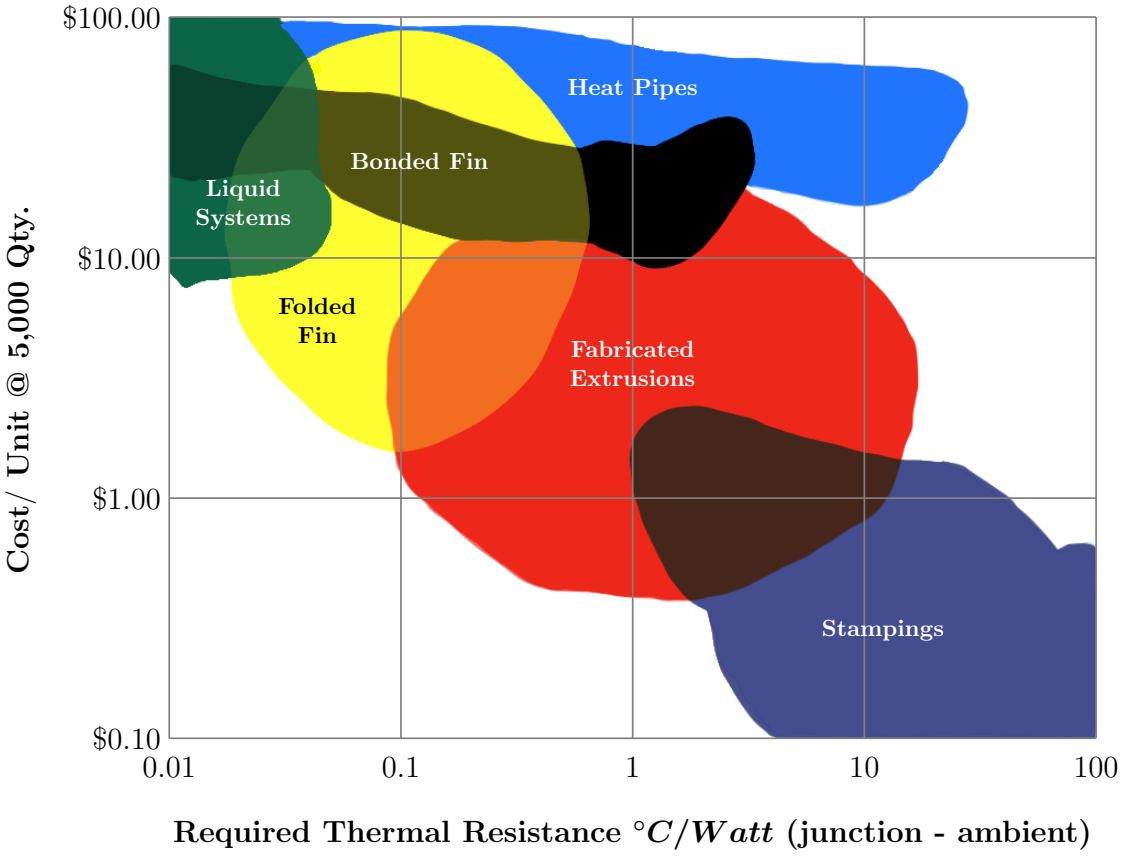


Figure 3.12: Relative Comparison of Various Types of Heat Sinks [83] [84]

The rise in cost of the different type of heat sinks is due to the manufacturing methods and their associated expense. This rise in cost per thermal resistance is nearly a linear trend and from that the assertion can be made that a lower required thermal resistance, results in a higher associated cost.

These cost analysis methods are not exhaustive in nature but, instead these methods will be used to offer additional characteristic comparisons of component choice. All of the aforementioned costing analysis will be applied in the results phase. The details of the application will be explained in Chapter 5.2.4 Results.

3.4 Energy Dissipation Circuit

All non-ideal circuits exhibit some parasitic energy storage. This energy is usually released during switch transitions. A robust design of a SSCB would account for the likely-hood that an energy transient occurs during operation. One solution is to install an energy dissipation circuit (EDC) in parallel to the switch. Any energy transient that occurs after the switching device opens would be stifled by this parallel branch. Thus, preventing any damage to the main conduction device. The most common cause of this energy transient is the high voltage pulse produced when direct current through an inductor is interrupted and is colloquially known as an inductive kickback.

As previously stated, an EDC can be made up of a single device, a series of devices, or any combination of device types in parallel. This EDC can be installed into a unidirectional circuit or bidirectional circuit, as seen in Figure 3.13 and Figure 3.14 respectively. The following discussion will focus on the operating challenges faced by an EDC, what energy dissipation device types are best suited for SSCBs, and how the selection and comparison criteria was developed.

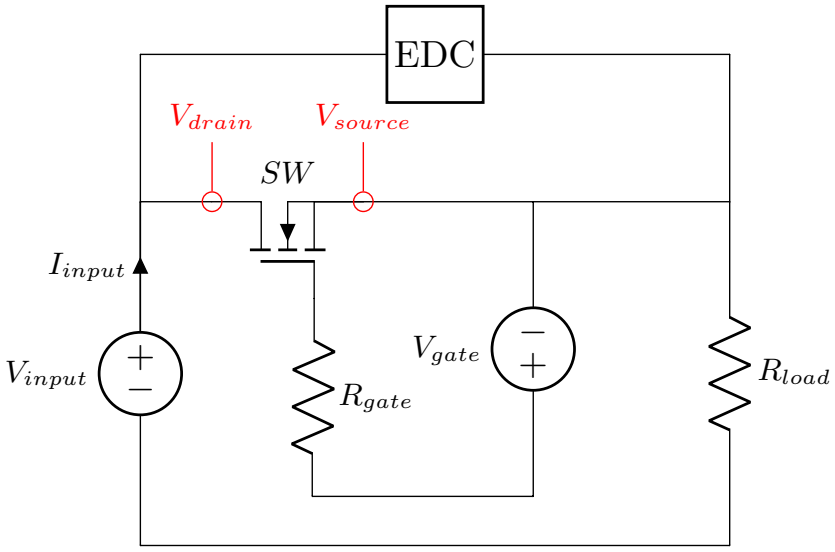


Figure 3.13: Unidirectional Solid-State Switch Circuit with Installed Energy Dissipation Circuit

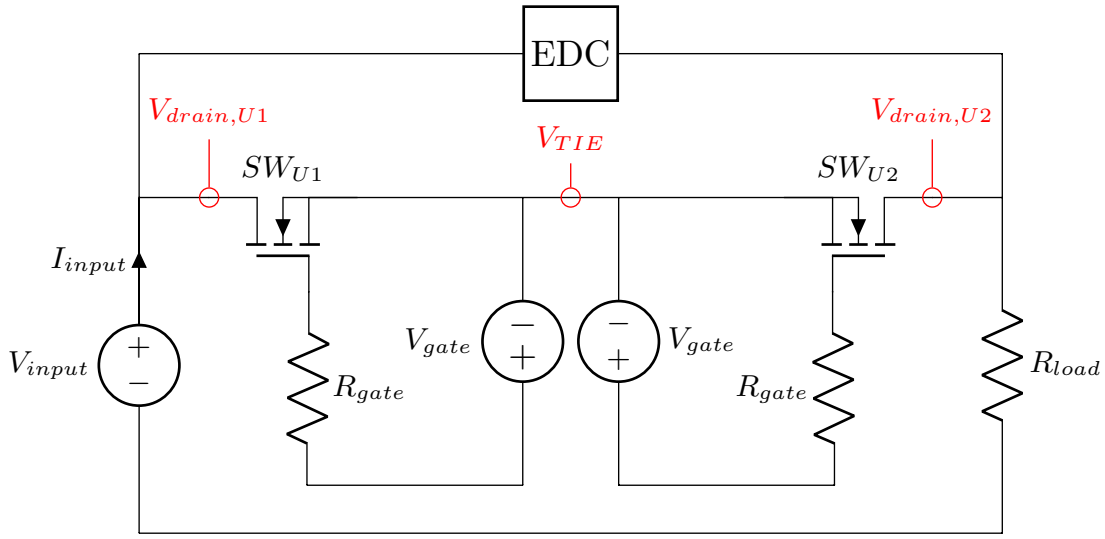


Figure 3.14: Bidirectional Solid-State Switch Circuit with Installed Energy Dissipation Circuit

3.4.1 Inductive Transient

Since all real world circuits have some inherent inductance, if not purposefully installed, the likely-hood of inductive transient occurring is high. This being the major operating challenge faced by an EDC. An inductive transient can also be considered a high voltage pulse, because current through an inductor can not change instantaneously. This can be demonstrated using the voltage and current equations of an inductor, Equation 3.18 and Equation 3.19 [3].

$$v = L\frac{di}{dt} \tag{3.18}$$

$$i = \frac{1}{L}\int_0^T vdt + i_0 \tag{3.19}$$

When a switch is opened under power with an inductive load attached, the current in the inductor would have to change to zero Amperes in zero seconds. Using Equation 3.18, the derivative of the current, di/dt, would be considered undefined or infinite. This would lead to the conclusion that there would be infinite voltage across the inductor, which cannot exist.

Since it has been determined that the current in an inductor cannot change instantaneously, this means an inductor is a stored energy device. This conclusion can be seen with the energy equation of an inductor, Equation 3.20 [3]. Since the inductance, (L, $Henry$), is a constant and the current, (I, $Ampere$), is greater than zero at the time of the switch opening, the energy in the inductor, (w, $Joule$), at the same time would also be greater than zero.

$$w = \frac{1}{2}LI^2 \tag{3.20}$$

The resultant energy stored in the inductor is a product of the magnetic flux density, (B). Using the previous energy equation, Equation 3.20, and a series of conversions, this can be demonstrated.

By using a solenoid as a representation of an inductor, in Equation 3.21 [76]. The magnetic flux of a solenoid or inductor, (Φ), is equal to the magnetic constant, (μ_0), the number of turns, (N), the current in the inductor, (i), and the cross-sectional area of the inductor, (A), over the length of the inductor, (l).

$$\Phi = \frac{\mu_0 N i A}{l} \tag{3.21}$$

Combining the previous magnetic flux equation, Equation 3.21, with the definition of inductance, Equation 3.22 [76]. Where inductance, (L), is equal to the number of turns in the inductor, (N), and the magnetic flux of the inductor, (Φ), over the current in the inductor, (i). A substitution for the inductance can be made in the previous energy equation, Equation 3.20.

$$\begin{aligned} L &= \frac{N\Phi}{i} \\ L &= \frac{\mu_0 N^2 A}{l} \end{aligned} \tag{3.22}$$

Using the definition of magnetic flux density, Equation 3.23 [76]. Where the magnetic flux density, (B), is equal to the magnetic constant, (μ_0), the number of turns in the inductor, (N), and the current in the inductor, (i), over the length of the inductor, (l).

Equation 3.23, can be rearranged to solve for current and substituted in the previous energy equation, Equation 3.20.

$$\begin{aligned} B &= \frac{\mu_0 N i}{l} \\ i^2 &= \frac{B^2 l^2}{\mu_0^2 N^2} \end{aligned} \tag{3.23}$$

A simplification of the substituted energy equation can be seen in Equation 3.24. In this equation the only attribute that varies in time is the magnetic flux density, (B). The magnetic constant, (μ_0), the cross-sectional area, (A), and the length of the inductor, (l), are all constant.

$$\begin{aligned} w &= \frac{1}{2} L i^2 \\ w &= \frac{1}{2} \frac{\mu_0 N^2 A}{l} \frac{B^2 l^2}{\mu_0^2 N^2} \\ w &= \frac{B^2}{2\mu_0} A l \end{aligned} \tag{3.24}$$

Having demonstrated the assumption that an inductor is an energy storage device and that the current of the inductor will not change instantly, this leads to the importance of an EDC.

More simply put, when any switch is opened under power with an inductive load, the inductor will generate a high potential across the switch. If a mechanical switch is being used this high voltage peak can cause a breakdown in the atmosphere and result in an electrical arc. If a solid-state switch is being used without an EDC the resultant high voltage peak will most likely damage the device rendering the switch inoperable.

3.4.2 Energy Dissipation Device

There are several devices capable of dissipating energy in a circuit each with their own advantages and disadvantages. Some options include MOVs, transient voltage suppressor diodes (TVS diodes), gas discharge tubes (GDTs), and thyristors. These options can be broken down into two distinct groups: clamping devices and crowbar devices.

A voltage clamping device is fully conducting at a fixed voltage magnitude. This attribute caps the voltage in the branch preventing any over voltage failures from occurring. A voltage crowbar device is similar to a clamping device in that there is a voltage level at which it is triggered, but does not hold or maintain the voltage at that level. It instead drives the voltage lower once triggered, usually to zero. This means power must be removed to stop the crowbar device from conducting.

When comparing performance, voltage clamping devices have faster response times over crowbar devices but are more limited in current conduction. This limitation is due to most of the energy of the transient is dissipated by the clamping device. This energy dissipation attribute is ideally suited for SSCB applications. Out of these two distinct groups, voltage clamping devices are better suited overall for SSCB applications.

3.4.2.1 Clamping Device Comparison

The voltage clamping devices under consideration are MOVs and TVS diodes. These devices are capable of withstanding the peak current transients that occur during SSCB operations. MOVs, also known as a metal-oxide varistors exhibit high resistance at low voltages that decrease as voltage across the device raises. TVS diodes, also known as a transient voltage suppressor diodes are similar to zener diodes in that they have a sharp operating knee, but are specifically designed and tested for voltage suppression.

When comparing voltage clamping devices some key characteristics are the voltage level at which the device starts conducting, the voltage level it is fully conducting, and the peak current capability. Other characteristics to consider are the reaction time and energy dissipation capacity, though these tend to be supplemental due to the voltage and current levels being more limiting.

The voltage level at which a clamping device is fully conducting is known as the clamp voltage. In MOVs it is abbreviated as V_C. In TVS diodes it is abbreviated as V_{CL}. The maximum voltage level a MOV can withstand without conducting is the V_M voltage and in TVS diodes it is known as the sustained operating voltage or V_{SO}. In between these two points is a voltage level at which a clamping device starts the bulk of its conduction. This mid-range voltage level is known as the breakover voltage. In MOVs the

breakover voltage is abbreviated as V_{NOM}. In TVS diodes the breakover voltage is abbreviated as V_{BR}.

The comparison of these clamping devices was conducted in two operations. First being a side by side comparison of all attributes centered around a common clamp voltage limit. Followed by an in depth view of the activation range of the same devices via a simulated circuit.

3.4.2.1.1 Voltage Clamping Attributes

Comparing the different clamping devices was completed by choosing a common clamp voltage limit. Determining this common clamp voltage limit, (V_{clamp}, V), was completed by using the standoff voltage limit of the solid state switch, (V_{DS}, V), plus twenty percent due to overshoot. This calculation can be seen in Equation 3.25. As previously noted, the clamp voltage limit may not be achievable with a single device and a series of devices may be necessary.

$$\begin{aligned} V_{clamp} &= V_{DS} \; + \; V_{DS} \; \times \; 0.2 \, [V] \\ 1440V &= 1200V + 240V \end{aligned} \tag{3.25}$$

A specific MOV model was chosen with a clamp voltage, (V_C, V), exceeding the previously defined common clamp voltage limit. Each of the relevant attributes of this MOV are displayed in Table 3.1, including the peak current limit, (I_{PK}, A).

Designation	**Model**	$\mathbf{V_{M(DC)}}$ (V)	$\mathbf{V_{NOM,MIN}}$ (V)	$\mathbf{V_{NOM,MAX}}$ (V)	$\mathbf{V_C}$ (V)	$\mathbf{I_{PK}}$ (A)
MOV	V575LA40A	730	805	1000	1500	50

Table 3.1: MOV Device Ratings and Specifications

Based on the above MOV clamp voltage, V_C, a specific TVS diode model was chosen. This model required two devices to be in series to match the specific clamp voltage limit. Each of the relevant attributes of this individual TVS diode model as well

as the resultant series combination are displayed in Table 3.2. Included in this table as well are the peak pulse current limits, (I_{PP}, A).

Designation	**Model**	$\mathbf{V_{SO}}$ (V)	$\mathbf{V_{BR,MIN}}$ (V)	$\mathbf{V_{BR,MAX}}$ (V)	$\mathbf{V_{CL}}$ (V)	$\mathbf{I_{PP}}$ (A)
TVS	AK10-530C	530	560	619	750	10,000
	(qty. 2 in series)	1060	1120	1238	1500	10,000

Table 3.2: TVS Diode Device Ratings and Specifications

In Table 3.1, in can be seen that the maximum standoff voltage of the MOV, V_M, is below the normal operating voltage of $1000V$ being considered in this thesis. This means that the MOV would be conducting during normal operations. Having the EDC continually conducting would drastically reduce the efficiency of the SSCB as a whole.

As can be seen in Table 3.2 the pulse peak current, I_{PP}, of the TVS diode far exceeds that of the MOV. This affords an EDC built with TVS diodes far greater capability in transient suppression.

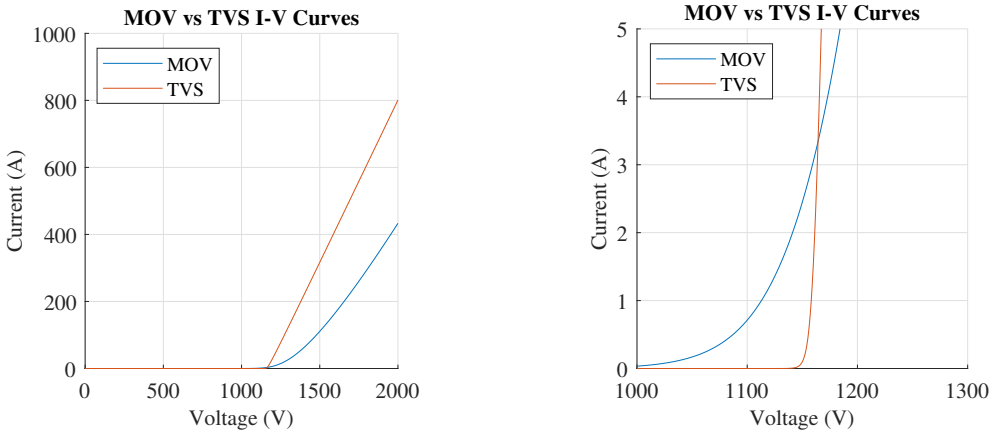
Finally, it can be seen that the MOV activation range, as defined by Equation 3.26, is nearly twice the size of the TVS diode series combination activation range. Specifically the MOV EDC activation range spans $770V$ when compared to the TVS diode EDC, which spans just $440V$. Further comparison the activation range will be completed in the following sub-section, 3.4.2.1.2: Voltage Clamping Activation Range.

$$\begin{aligned} V_{active,MOV} &= V_C \ - \ V_{M(DC)} \, [V] \\ V_{active,TVS} &= V_{CL} \ - \ V_{SO} \, [V] \end{aligned} \tag{3.26}$$

3.4.2.1.2 Voltage Clamping Activation Range

The TVS diode EDC has a sharper activation knee, this can be more clearly seen in Figure 3.15. In this figure the full range of each of EDC is shown side-by-side with a more detailed view of the activation range. In Figure 3.15a, the voltage axis is from $0V$ to $2000V$, well above the rated clamping voltage of $1500V$. In Figure 3.15b, the same data is

displayed but, the voltage and current axes are limited to show a more detailed view of the activation range. Here, the sharp knee of the TVS diode EDC can be seen.



(a) MOV vs. TVS Diode Full Range (b) MOV vs. TVS Diode Activation Range

Figure 3.15: MOV and TVS Diode I-V Curve Comparison

These results yield that TVS diodes EDC warrant more investigation into specific models. This investigation will be completed in Chapter 4.5.3 Simulation Setup and Test, using the criteria described in the following sub-sections 3.4.3: Selection Criteria and 3.4.4: Comparison Criteria.

The above current-voltage (I-V) curves were achieved using simulated circuits in LTspice, as seen in Figure 3.16 and Figure 3.17. In these test circuits, a DC voltage source was swept from $0V$ to $2000V$ at $1V$ increments. The clamping device was placed in series with the voltage source and a resistive load of 1Ω. The load only affects the final current of the transient sweep.

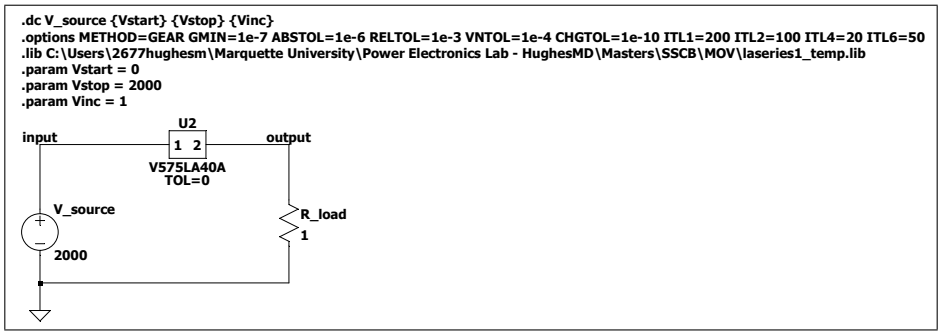


Figure 3.16: MOV I-V Curve Circuit Simulations with Littelfuse V575LA40A

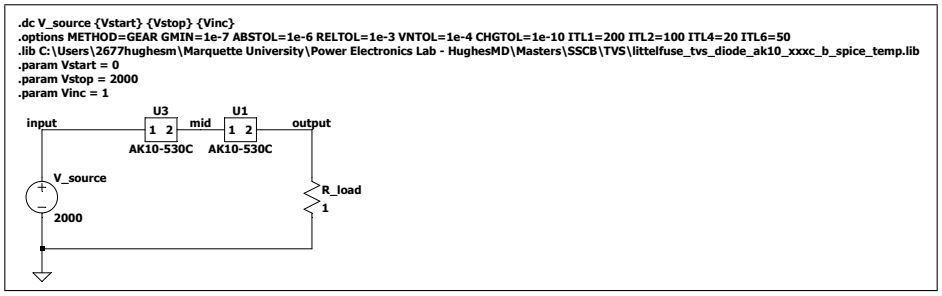


Figure 3.17: TVS Diode I-V Curve Circuit Simulations with Littelfuse AK10530C

LTspice produces a ".raw" file, which contains arrays of voltages at each node and arrays of currents of each component. These voltage and current arrays are associated with an included time array. This ".raw" file was exported as a plain text file and imported in MATLAB. In MATLAB the voltage difference from the input node and output node was used as the x-axis. Whereas the current into each clamping device was used as the y-axis. The result of this process can be seen in Figure 3.15.

3.4.3 Selection Criteria

The following sections will describe the criteria to be used in the selection process between TVS diodes. A single TVS diode can not achieve the necessary stand off voltage limit for the purposes of this thesis. Therefore, series combinations of TVS diodes will be considered to achieve the necessary withstand voltage limit. Many TVS diodes have higher than necessary peak current limits for the purposes of this thesis. Therefore, there is no need for parallel combinations within the EDC with respect to this thesis, though this capability does exist.

The selection process can be seen in Figure 3.18. As depicted the process will be used to down select between TVS diodes that are readily available from electronic component distributors. After which a series of simulations will be completed using the devices. The simulations will generate results for comparison.

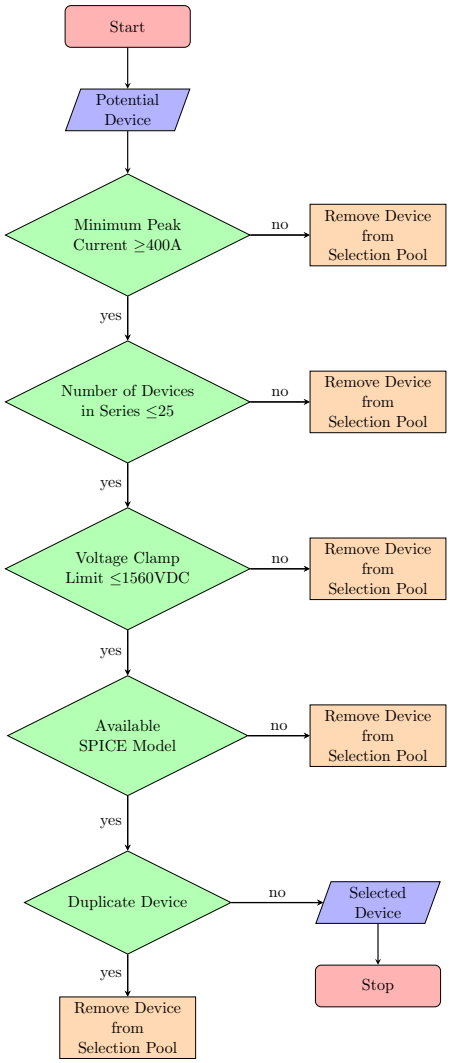


Figure 3.18: Energy Dissipation Circuit Device Selection Flowchart

3.4.3.1 Pulse Peak Current

Since parallel branches within the EDC will not be considered, the first criteria used for down selection would be the pulse peak current, (I_{PP}, A). The minimum pulse peak current limit, ($I_{PP,MIN}$, A), was derived by calculating the peak current during a shorting event with a ratio of 10 and with an added margin of twenty percent of that peak current. This calculation resulted in a peak current limit of $400A$.

$$\begin{aligned} I_{PP,MIN} &= \frac{V_{input}}{R_{short}} + \left(\frac{V_{input}}{R_{short}}\right) \times 0.2\,[A] \\ 400A &= \frac{1000V}{3\Omega} + \left(\frac{1000V}{3\Omega}\right) \times 0.2 \end{aligned} \tag{3.27}$$

The above derivation was completed using the input supply voltage, (V_{input}, V), and the shorted resistance value, (R_{short}, Ω). The derivation of the shorted resistance value can be seen in Equation 3.28, where the load resistance, (R_{load}, Ω), is used in conjunction with the shorting ratio.

$$\begin{aligned} 10 &= \frac{R_{load}}{R_{short}} \\ R_{short} &= R_{load} \times 0.1\ [\Omega] \\ 3\Omega &= 30\Omega \times 0.1 \end{aligned} \tag{3.28}$$

In some cases the pulse peak current limit, (I_{PP}, A), is not listed on the manufacturer provided data sheet. In these select cases the following equations will be used to determine the pulse peak current limit. Equation 3.29 will use the provided pulse peak power limit, (P_{PP}, kW) and the clamp voltage limit, (V_{CL}, V). If no clamp voltage limit is provided then the maximum break over voltage, ($V_{BR,MAX}$, V), will be used as seen in Equation 3.30.

$$I_{PP,clamp} = \frac{P_{PP}}{V_{CL}}\ [A] \tag{3.29}$$

$$I_{PP,breakover} = \frac{P_{PP}}{V_{BR,MAX}}\ [A] \tag{3.30}$$

3.4.3.2 Series Number of Devices

Determining the number of devices in series necessary for the EDC to achieve the minimum desired stand off voltage can be achieved using Equation 3.31. Wherein the input supply voltage, (V_{input}, V) is divided by the stand off voltage limit, (V_{SO}, V), of the device resulting in an integer rounded up to the nearest whole number, (n_{EDC}). The number of allowable devices in series will be limited to no more than 25. This will reduce any potential manufacturing errors or errors in simulation that may be magnified by the number of devices.

$$n_{EDC} = \lceil \frac{V_{input}}{V_{SO}} \rceil \tag{3.31}$$

If there is no stand off voltage limit listed on the manufacturer's data sheet then Equation 3.32 will be used. Here the stand off voltage of the solid state switch, (V_{DS}, V), which is greater than the input supply voltage, is divided by the break over voltage limit, $V_{BR,MAX}$ of the TVS diode.

$$n_{EDC} = \frac{V_{DS}}{V_{BR,MAX}} \tag{3.32}$$

3.4.3.3 Clamping Voltage Limit of the EDC

The next step in down selection would be to calculate the clamping voltage of the EDC and limit all candidates that exceed the allowable clamping voltage limit. This limit was previously defined as $1440V$ using Equation 3.25. The clamping voltage of the entire circuit, ($V_{CL,EDC}$, V), can be determined using the required number of devices in series, (n_{EDC}), and the individual clamping limit, (V_{CL}, V) as seen in Equation 3.33.

$$V_{CL,EDC} = V_{CL} \times n_{EDC}\,[V] \tag{3.33}$$

The previously defined allowable clamping voltage limit of $1440V$ accounted for an overshoot of twenty percent. This limit may be to restricting and a more lenient voltage overshoot of thirty percent may need to be considered, resulting in a maximum allowable clamping voltage limit of $1560V$, as seen in Equation 3.34. If a decision is made to use this new maximum limit the body-diode standoff voltage must be taken into consideration. If the clamping voltage limit is too high, then the body-diode will become reversed biased and conduct current. This will bypass the EDC branch all together, putting equipment and personnel at risk.

$$\begin{aligned} V_{clamp,MAX} &= V_{DS} \; + \; V_{DS} \; \times \; 0.3\,[V] \\ 1560V &= 1200V + 360V \end{aligned} \tag{3.34}$$

3.4.3.4 Modeling of the EDC

Finally, the selection pool of devices will be surveyed for available SPICE models. These device models will be used in the simulation phase of this thesis. If no SPICE model is available for a particular device it will be removed from consideration. Additionally, any duplicate models that may have arisen due to manufacturer model numbering intricacies will also be removed from the selection pool.

3.4.3.5 Additional Attributes of the EDC

The operating limits of the whole EDC can be determined using individual device characteristics in conjunction with the number of required devices in series, (n_{EDC}). These limits include the stand off voltage ($V_{SO,EDC}$, V), the minimum break over voltage ($V_{BR,MIN,EDC}$, V), and the maximum break over voltage ($V_{BR,MAX,EDC}$, V). Using Equation 3.35, Equation 3.36, and Equation 3.37 respectively.

$$V_{SO,EDC} = V_{SO} \times n_{EDC}\,[V] \tag{3.35}$$

$$V_{BR,EDC,MIN} = V_{BR,MIN} \times n_{EDC}\,[V] \tag{3.36}$$

$$V_{BR,EDC,MAX} = V_{BR,MAX} \times n_{EDC}\,[V] \tag{3.37}$$

3.4.4 Comparison Criteria

Once a final selection pool has been determined a set of comparison criteria will applied. This comparison criteria will take into account the total cost of the EDC, cost per power capacity of individual devices, the activation range of the EDC, as well as the clamping performance of the EDC as a whole. In regards to the listed comparison criteria: a lower total cost of the EDC, lower cost per power capacity, and a narrow activation region is desirable. Additionally, quicker response times and higher energy absorption after transients would be signs of a well performing EDC.

3.4.4.1 Cost

The total cost of the EDC, ($Price_{total}$, \$), will be determined using the number of devices required in series, (n_{EDC}), and the cost of an individual device, ($Price_{unit}$, \$). Equation 3.38 displays this determination.

$$Price_{total} = n_{EDC} \times Price_{unit} \, [\$] \tag{3.38}$$

3.4.4.2 Power

The cost per power capacity, ($Price_{watt}$, $\$/kW$), can be determined by applying Equation 3.39. Here the cost an individual device, ($Price_{unit}$, \$), is divided by the peak pulse power limit. The peak pulse power limit, (P_{PP}, kW), used would be provided on the manufacturer data sheet.

$$Price_{watt} = Price_{unit}/P_{PP} \, [\$/kW] \tag{3.39}$$

If the pulse peak power limit, P_{PP}, is not listed on the manufacturer provided data sheet, it will be determined by Equation 3.40. Here the pulse peak current limit, I_{PP}, and the individual device clamp voltage limit, (V_{CL}, V), is used in conjunction with the power law equation from Chapter 1.3, Equation 1.3.

$$P_{PP,calculated} = I_{PP} \times V_{CL} \, [kW] \tag{3.40}$$

3.4.4.3 Activation Range

Next, the voltage activation range of the EDC, ($V_{active,EDC}$, V), will be determined as seen in Equation 3.41. This derivation will be completed using the previously derived EDC voltage clamping limit in Equation 3.33 and EDC voltage stand

off limit in Equation 3.35.

$$V_{active,EDC} = V_{CL,EDC} - V_{SO,EDC}\,[V] \tag{3.41}$$

3.4.4.4 Clamping Performance

Finally, the simulated clamping performance of each EDC will be evaluated for turn-on voltage, peak voltage, and the time duration between these values. Comparing these voltage values to the derived EDC voltage limits will bring insight in to the accuracy of the manufacturer reported data and the manufacturer provided simulation model. The details of this comparison will be explained in Chapter 4.5.3 Simulation Setup and Test.

An additional aspect of comparison between the EDCs will be in terms of energy absorption. This evaluation will be completed during a voltage clamping transient. If all other aspects of the simulated circuit are equal, then the EDC with the highest energy absorbed will provide more benefits in terms of switch circuit longevity.

3.4.4.5 Additional Attributes for Comparison

Another attribute to consider adjacent to the activation range is the breaking voltage range. The voltage breaking range of the EDC, ($V_{break,EDC}$, V), will be determined as seen in Equation 3.42. This derivation will be completed using the previously derived EDC maximum breaking voltage in Equation 3.37 and EDC minimum breaking voltage in Equation 3.36.

$$V_{break,EDC} = V_{BR,MAX,EDC} - V_{BR,MIN,EDC}\,[V] \tag{3.42}$$

The aforementioned selection and comparison analysis will be completed in the simulation phase of this thesis. As previously mentioned, the details of the application will be explained in Chapter 4.5.3 Simulation Setup and Test.

CHAPTER 4
SIMULATION SETUP AND TEST

The following simulations will fully realize the previously discussed analytical reasoning. This will allow for the informed decision-making of primary components. The simulation phase will be explained and applied to a unidirectional circuit as well as a bidirectional circuit. Any limiting parameters will be explained as needed.

4.1 Solid-State Switch Simulation Format

A list of eleven solid-state switch models from six different manufacturers were simulated. This list was generated from readily available solid-state switches from electronic component distributors. The chosen solid-state switches had to meet the minimum blocking voltage, (V_{DS}, $volt$), set forth in the Introduction, Chapter 1.3. Additionally, the group of devices had to possess similar on-state resistances, ($R_{DS(ON)}$, $m\Omega$).

In an attempt to diversify the field of solid-state switches to be simulated, JFETs, MOSFETs, co-packaged cascode configurations of JFETs and MOSFETs, as well IGBTs were simulated. The JFETs simulated were used as a benchmark of on-state resistance for all other solid-state switches simulated. The complete list of simulated solid-state switch models is shown below in Table 4.1 [72] [74] [85] [86] [87] [88] [89] [73] [90] [75] [91].

Designation	**Manufacturer**	**Model**	**Description**	$\mathbf{V_{DS/CE}}$ (V)	$\mathbf{I_{D/C}}$ (A)	$\mathbf{R_{(ON)}}$ ($m\Omega$)	$\mathbf{T_J}$ ($^\circ C$)
Device 1	Infineon	IMW120R060M1H	MOSFET	1200	36	60	175
Device 2	Littelfuse	LSIC1MO120E0080	MOSFET	1200	39	80	150[2]
Device 3	Microchip	MSCO80SMA120J	MOSFET	1200	37	80	175
Device 4	Rohm	SCT3080KLGC11	MOSFET	1200	31[1]	80	175
Device 5	United SiC	UJ3N120070K3S	JFET	1200	33.5	70	175
Device 6	United SiC	UJ3C120080K3S	Cascode	1200	33[1]	80	175
Device 7	United SiC	UJ3C120070K3S	Cascode	1200	34.5	70	175
Device 8	Wolfspeed	C2M0080120D	MOSFET	1200	36	80	150[2]
Device 9	Wolfspeed	C3M0075120D	MOSFET	1200	30[1]	75	150[2]
Device 10	Infineon	IKW15N120BH6	IGBT	1200	30[1]	91.7[3]	175
Device 11	Infineon	IKW15N120T2	IGBT	1200	30[1]	79.2[3]	175

Note 1: simulations will exceed stated manufacturer drain current limits at 25°C
Note 2: simulations will exceed stated manufacturer junction temperature limits
Note 3: on-state resistance has been estimated using manufacturer provided data

Table 4.1: Solid-State Switch Primary Attributes

The chosen devices will be simulated operating at an ambient temperature or case temperature of $25°C$ with various junction temperatures. The manufacturer stated maximum junction temperature of each device is not the same. The chosen Littelfuse Inc. model as well as the chosen Wolfspeed models have a junction temperature upper limit of $150°C$, which is in contrast to all other models chosen that have an upper limit of $175°C$.

To ensure consistent comparisons between the simulated operations, all models will be simulated at the same temperatures regardless of manufacturers stated limits. The results from the most limiting junction temperature will also be included in the results. This will highlight any drastic changes in operation at higher operating temperatures and will allow for a more informed component decision.

Along with operating temperatures, the chosen devices will be simulated operating at various loads. These load values will drive the on-state currents, with a minimum of $10A$ and a maximum of $33\frac{1}{3}A$. This chosen maximum exceeds some of the manufacturer stated continuous on-state current limits. The following models will be simulated past their stated limit for a portion of the data points collected: Device 4, Device 6, Device 9, Device 10, and Device 11.

The discrete data points from the most limiting continuous on-state current parameter will also be included in the results. This lower load point of 34Ω will have a continuous on-state current of approximately $29.4A$. This is less than the most limited switch parameter of $30A$ of Device 9, Device 10, and Device 11. Much like with the junction temperature, this will highlight any drastic changes in operation at higher drain currents and will allow for a more informed component decision.

4.2 Switch Efficiency Simulation

The efficiency of each device was determined by using a combination of software, being MATLAB and LTspice. MATLAB, which is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks, having been the primary tool used for setup and calculations. LTspice, which is a SPICE-based analog electronic circuit simulator computer software, produced by semiconductor manufacturer Analog Devices, was used for simulation solving and data generation. The

MATLAB code used for the simulations can be found in full in the Appendix C: Software Code, along with all other software code written for this thesis.

All efficiency circuits whether unidirectional or bidirectional were simulated with an input voltage of $1kV$. The simulated load was purely resistive and iterated from 30Ω to 100Ω at 1Ω increments. Additionally, each load iteration cycle was simulated at four junction temperatures being, -55°C, 25°C, 150°C, and 175°C. These temperatures represent the low end limit, ambient, limited high end, and high end limit respectively of most solid-state switch operating temperatures. The combination of load and temperature iterations resulted in 284 data points for each solid-state switch simulated.

4.2.1 Simulation Setup Description

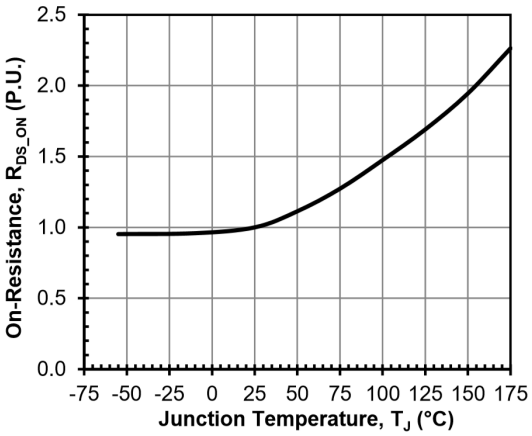
The simulation setup consisted of identifying and defining all script-able simulation and SPICE circuit attributes. Some items were straight forward, such as, simulation run time and static component values. Other items required a more nuanced approach, for instance, junction temperature, (T_j). A variety of methods were required to control the junction temperature of each solid-state switch model, depending on the manufacturer.

Some manufacturer provided SPICE models allowed for the direct control of the junction temperature, (T_j). This was completed by either a direct parameter input or by an input voltage pin. Using an independent voltage source connected to the input pin, as directed by the manufacturer data sheet, the output voltage of that source would directly correlate to the desired temperature. Those provided models included: Device 1, Device 2, Device 3, Device 8, Device 9, Device 10, and Device 11.

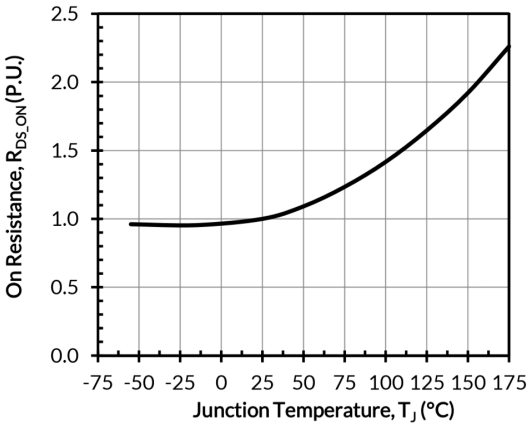
Device 4 was not provided a SPICE model with direct control via an input pin, but rather allowed the model file to be edited to control the junction temperature, (T_j). This was accomplished by using MATLAB to open the model file, which was in plain text format, edit the temperature parameter and then run the simulation for each desired temperature point.

Manufacturers for Device 5, Device 6, and Device 7 also did not provide a SPICE model with direct control via an input pin nor did they provide a direct text input option into the model file. Instead, the on-resistance was modified to achieve each junction

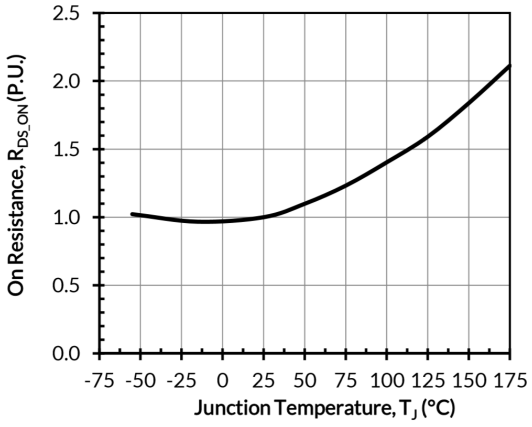
temperature. First, the normalized on-resistance, ($R_{DS(ON)}$), versus junction temperature, (T_j), profile from the manufacturer provided data sheet was approximated. This approximation was used to generate corresponding on-resistances from the desired junction temperature. Then the plain text SPICE model file was edited via MATLAB, similar to how it was accomplished with the Device 4 SPICE model. The manufacturer provided data sheet temperature profiles can be seen in Figures 4.1a, 4.1b, and 4.1c, with the corresponding approximation in Figure 4.1d. The same approximation profile was used for each SPICE model of Device 5, Device 6, and Device 7. Generating more defined temperature profiles for each of the switch models would be an appropriate subject matter for future work.



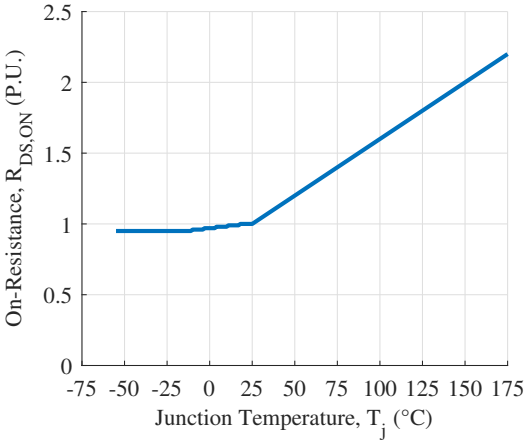
(a) Device 5 at $V_{GS} = 0V$ and $I_D = 10A$



(b) Device 6 at $V_{GS} = 12V$ and $I_D = 20A$



(c) Device 7 at $V_{GS} = 12V$ and $I_D = 20A$



(d) On-State Resistance versus Junction Temperature Generated Approximation

Figure 4.1: Normalized On-State Resistance vs. Junction Temperatures

4.2.2 SPICE Circuit Description

The SPICE circuit section of MATLAB code consisted of defining components in a nodal matrix, called a netlist [92]. During each simulation MATLAB would open the SPICE netlist file, edit the file accordingly, and then run the simulation. The simulation was completed by LTspice, but the initiation of the simulation was controlled by MATLAB. The data was later collected by MATLAB via an import tool. This process will be covered in more detail below in the Calculation Description Subsection.

Ensuring each simulation circuit netlist was editable by MATLAB was key to the script-able attribute of the software code. This allowed for the iteration of load values and junction temperatures, as well as allowing for future work in optimization which will be covered in more detail in Conclusions, Chapter 6.6. An example of the resultant circuit created from one of the simulated circuit netlists can be seen in Figure 4.2. This example is of the Device 9 unidirectional efficiency simulation and which can be found in full in the Appendix B: Schematics, along with all other generated simulation circuits for this thesis.

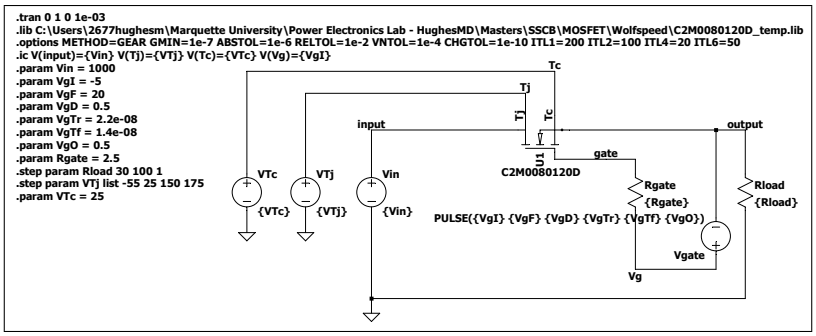


Figure 4.2: Unidirectional Efficiency of Device 9

In addition to the defining the component values of the simulated circuit, attributes controlling how LTspice solved the circuit were also defined. These modified options remained consistent between all simulated circuits. Below is listed all options that were changed from a default state, what their purpose is, what their original value was, and their newly defined value. All of these values were changed to help with convergence, reduce simulation time duration, or ensure the completion of the simulation. Available future work would be to define the accuracy options more closely to the default values.

- Numerical Integration Options
  - METHOD: selects the numerical integration formula; the default is the second-order trapezoidal method.
    - METHOD = $GEAR$
  - CHGTOL: the absolute charge tolerance at any time point; the default is $1 \times 10^{-14} C$.
    - CHGTOL = 1×10^{-10}
- Linear Equation Options
  - GMIN: defines the minimum conductance connected in parallel to a pn junction; the default is $1 \times 10^{-12} mho$.
    - GMIN = 1×10^{-7}
- Nonlinear Solution Options
  - ABSTOL: represents the absolute current tolerance. The smallest current that can be monitored is equal to value; the default is $1 \times 10^{-12} A$.
    - ABSTOL = 1×10^{-6}
  - RELTOL: defines the relative error tolerance within which voltages and device currents are required to converge; the default is 1×10^{-3}.
    - RELTOL = 1×10^{-2}
  - VNTOL: the absolute voltage tolerance. It represents the smallest observable voltage; the default is $1 \times 10^{-6} V$.
    - VNTOL = 1×10^{-4}
  - ITL1: sets the maximum number of iterations used for the DC solution; the default is 100.
    - ITL1 = 200
  - ITL2: sets the number of iterations allowed for any new source value; the default is 50.
    - ITL2 = 100
  - ITL4: sets an upper limit to the number of iterations performed at a time point before it is rejected and the time step reduced by 8; the default is 10.
    - ITL4 = 20
  - ITL6: represents both a flag for source ramping in a DC solution and the maximum number of iterations allowed for each stepped value of the supplies; the default is 25.
    - ITL6 = 50

Again, all of the options modified were for the purpose of assisting the solver complete the simulation in a timely manner. Some of the options have reduced the overall accuracy of the results. Not all simulated circuits required these options over all temperature ranges, or at all. In an effort to maintain any inaccuracies relative, the same options were used across the testing pool.

4.2.3 Simulation Switch Settings

When simulating each of the solid-state switches, besides the simulation options, four attributes were key to ensuring the simulation was completed over the entire load range for each junction temperature. The gate-to-source voltage (V_{GS},V) or in the case of IGBTs the gate-to-emitter voltage (V_{GE},V), external gate resistance ($R_{G,ext}$,Ω), rise time (t_r,ns), and fall time (t_f,ns) all played a role in ensuring the simulation was completed in a timely manner. These attributes were commonly, but not always, defined in the manufacturer's data sheet. The following tables will list the typical and maximum values if they were defined. Also, if present any testing conditions will be listed. As a reminder, solid-state switch models with recommended external gate resistance of 0Ω had no external gate resistance installed.

The simulations conducted in this thesis fall out of typical conditions, therefore some of these attributes required adjusting. At any point the attributes were adjusted a comment was made as to reasoning or thought process. It is worth noting that the independent voltage source applied to the gate pin, also known as the gate-to-source voltage, had the largest affect on operation and efficiency. Just by raising this voltage one could artificially boost efficiency, so it was key to maintain this voltage as close to typical as possible.

4.2.3.1 Device 1

A negative gate voltage was used to further drive the off-state of the switch in both the unidirectional and bidirectional circuit. This setting was only necessary in the highest junction temperature range of $175°C$. Each of the settings used, including the manufacturer's recommended values can be seen in Table 4.2.

Device 1	**Typical**	**Max**	**Manufacturer Test Conditions**	**Unidirectional Simulation**	**Bidirectional Simulation**
$V_{GS}(V)$	0/+15...18	-7/+23	0/+18	-0.1/+18	-0.1/+18
$R_{G,ext}(\Omega)$	-	-	2	2	2
$t_r(ns)$	17	-	-	17	17
$t_f(ns)$	12	-	-	12	12

Table 4.2: Device 1 Switch Settings [86]

4.2.3.2 Device 2

No changes from the manufacturer's testing condition were necessary to operate this switch in the unidirectional configuration over the entire load range for each junction temperature. The bidirectional circuit required a deviation from the recommended external gate resistance value. During simulation, LTspice would error citing an issue with the time step size being too small. Reducing the external gate resistance of both switches prevented this error from occurring. Reducing this resistance changed the charging time constant of the switch. Changing this time constant has no impact on the steady state operation of the switch. Each of the settings used, including the manufacturer's recommended values can be seen in Table 4.3.

Device 2	**Typical**	**Max**	**Manufacturer Test Conditions**	**Unidirectional Simulation**	**Bidirectional Simulation**
$V_{GS}(V)$	-5/+20	-6/+22	-5/20	-5/+20	-5/+20
$R_{G,ext}(\Omega)$	-	-	2	2	1.9
$t_r(ns)$	10	-	-	10	10
$t_f(ns)$	6	-	-	6	6

Table 4.3: Device 2 Switch Settings [87]

4.2.3.3 Device 3

No changes from the manufacturer's testing condition were necessary to operate this switch over the entire load range for each junction temperature in either the unidirectional or bidirectional circuit. Each of the settings used, including the manufacturer's recommended values can be seen in Table 4.4.

Device 3	**Typical**	**Max**	**Manufacturer Test Conditions**	**Unidirectional Simulation**	**Bidirectional Simulation**
$V_{GS}(V)$	-	-10/+23	-5/+20	-5/+20	-5/+20
$R_{G,ext}(\Omega)$	-	-	4	4	4
$t_r(ns)$	4	-	-	4	4
$t_f(ns)$	15	-	-	15	15

Table 4.4: Device 3 Switch Settings [88]

4.2.3.4 Device 4

No changes from the manufacturer's testing condition were necessary to operate this switch over the entire load range for each junction temperature in either the unidirectional or bidirectional circuit. Each of the settings used, including the manufacturer's recommended values can be seen in Table 4.5.

Device 4GC11	**Typical**	**Max**	**Manufacturer Test Conditions**	**Unidirectional Simulation**	**Bidirectional Simulation**
$V_{GS}(V)$	0/+18	-4/+22	0/+18	0/+18	0/+18
$R_{G,ext}(\Omega)$	-	-	0	0	0
$t_r(ns)$	15	-	-	15	15
$t_f(ns)$	24	-	-	24	24

Table 4.5: Device 4 Switch Settings [89]

4.2.3.5 Device 5

No changes from the manufacturer's testing condition were necessary to operate this switch over the entire load range for each junction temperature in either the unidirectional or bidirectional circuit. Each of the settings used, including the manufacturer's recommended values can be seen in Table 4.6.

Device 5	**Typical**	**Max**	**Manufacturer Test Conditions**	**Unidirectional Simulation**	**Bidirectional Simulation**
$V_{GS}(V)$	-	-20/+3	-18/0	-18/0	-18/0
$R_{G,ext}(\Omega)$	-	-	1	1	1
$t_r(ns)$	25	-	-	25	25
$t_f(ns)$	39	-	-	39	39

Table 4.6: Device 5 Switch Settings [72]

4.2.3.6 Device 6

No changes from the manufacturer's testing condition were necessary to operate this switch over the entire load range for each junction temperature in either the

unidirectional or bidirectional circuit. Each of the settings used, including the manufacturer's recommended values can be seen in Table 4.7.

Device 6	**Typical**	**Max**	**Manufacturer Test Conditions**	**Unidirectional Simulation**	**Bidirectional Simulation**
$V_{GS}(V)$	-5/+15	-25/+25	-5/+15	-5/+15	-5/+15
$R_{G,ext}(\Omega)$	-	-	1/20[1]	1	1
$t_r(ns)$	14	-	-	14	14
$t_f(ns)$	14	-	-	14	14

Note 1: Manufacturer Test Conditions listed Turn-on $R_{G,ext}$ as 1Ω and Turn-off $R_{G,ext}$ as 20Ω.

Table 4.7: Device 6 Switch Settings [74]

4.2.3.7 Device 7

No changes from the manufacturer's testing condition were necessary to operate this switch over the entire load range for each junction temperature in either the unidirectional or bidirectional circuit. Each of the settings used, including the manufacturer's recommended values can be seen in Table 4.8.

Device 7	**Typical**	**Max**	**Manufacturer Test Conditions**	**Unidirectional Simulation**	**Bidirectional Simulation**
$V_{GS}(V)$	-5/+15	-25/+25	-5/+15	-5/+15	-5/+15
$R_{G,ext}(\Omega)$	-	-	1/22[1]	1	1
$t_r(ns)$	17	-	-	17	17
$t_f(ns)$	9	-	-	9	9

Note 1: Manufacturer Test Conditions listed Turn-on $R_{G,ext}$ as 1Ω and Turn-off $R_{G,ext}$ as 22Ω.

Table 4.8: Device 7 Switch Settings [85]

4.2.3.8 Device 8

No changes from the manufacturer's testing condition were necessary to operate this switch over the entire load range for each junction temperature in either the unidirectional or bidirectional circuit. Each of the settings used, including the manufacturer's recommended values can be seen in Table 4.9.

Device 8	Typical	Max	Manufacturer Test Conditions	Unidirectional Simulation	Bidirectional Simulation
$V_{GS}(V)$	-5/+20	-10/25	-5/+20	-5/+20	-5/+20
$R_{G,ext}(\Omega)$	-	-	2.5	2.5	2.5
$t_r(ns)$	22	-	-	22	22
$t_f(ns)$	14	-	-	14	14

Table 4.9: Device 8 Switch Settings [73]

4.2.3.9 Device 9

No changes from the manufacturer's testing condition were necessary to operate this switch over the entire load range for each junction temperature in either the unidirectional or bidirectional circuit. It is worth noting that almost all relaxed simulation options were driven from this one switch configuration. The bidirectional circuit had the longest simulation completion time out of the entire selection pool, by several orders of magnitude. The simulation duration could be shortened by increasing the gate voltage. Doing so however would artificially boost the efficiency of the circuit. This switch simulation would be a prime candidate for future work. Each of the settings used, including the manufacturer's recommended values can be seen in Table 4.10.

Device 9	Typical	Max	Manufacturer Test Conditions	Unidirectional Simulation	Bidirectional Simulation
$V_{GS}(V)$	-4/+15	-8/+19	-4/+15	-4/+15	-4/+15
$R_{G,ext}(\Omega)$	-	-	0	0	0
$t_r(ns)$	17	-	-	17	17
$t_f(ns)$	13	-	-	13	13

Table 4.10: Device 9 Switch Settings [90]

4.2.3.10 Device 10

No changes from the manufacturer's testing condition were necessary to operate this switch over the entire load range for each junction temperature in either the unidirectional or bidirectional circuit. Each of the settings used, including the manufacturer's recommended values can be seen in Table 4.11.

Device 10	Typical	Max	Manufacturer Test Conditions	Unidirectional Simulation	Bidirectional Simulation
$V_{GE}(V)$	-	-20/+20	0/+15	0/+15	0/+15
$R_{G,ext}(\Omega)$	-	-	22	22	22
$t_r(ns)$	29	-	-	29	29
$t_f(ns)$	25	-	-	25	25

Table 4.11: Device 10 Switch Settings [75]

4.2.3.11 Device 11

No changes from the manufacturer's testing condition were necessary to operate this switch over the entire load range for each junction temperature in either the unidirectional or bidirectional circuit. Each of the settings used, including the manufacturer's recommended values can be seen in Table 4.12.

Device 11	Typical	Max	Manufacturer Test Conditions	Unidirectional Simulation	Bidirectional Simulation
$V_{GE}(V)$	-	-20/+20	0/+15	0/+15	0/+15
$R_{G,ext}(\Omega)$	-	-	41.8	41.8	41.8
$t_r(ns)$	25	-	-	25	25
$t_f(ns)$	95	-	-	95	95

Table 4.12: Device 11 Switch Settings [91]

4.2.4 Calculation Description

LTspice is capable of performing calculations internal to it's own environment and producing waveforms that are exportable as an enhanced windows media file, also known as a ".emf" file. Though for the purposes of this thesis, MATLAB's environment was more flexible and robust. The data collection and calculation portion of the MATLAB code included using the function, "LTspice2Matlab" to import the ".raw" data file from each LTspice simulation. However, LTspice was used to ensure expected operation of each circuit configuration over the full range of testing conditions.

This assurance was completed with a cursory view of the LTspice waveforms. This included looking for any abnormalities in the output voltage, such as early turn off. An example of the output voltage waveforms that were reviewed can be seen in Figure 4.3.

This example is from the Device 9 bidirectional efficiency simulation of all 284 operating conditions at the turn-on moment of the switch, specifically $500ms$. As can be seen in Figure 4.3, discerning between operating conditions or generating discrete points would be difficult, thus reiterating the necessity to import the waveform data into MATLAB.

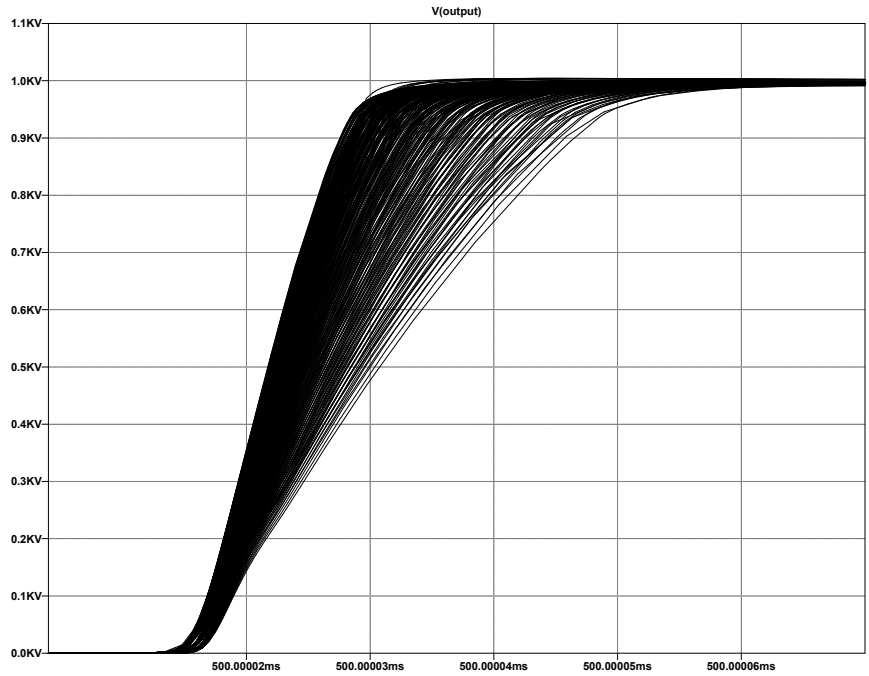


Figure 4.3: Device 9 Bidirectional Efficiency Simulation Output Voltage Example Waveform at Switch Turn-On with 284 Testing Conditions Displayed

The imported ".raw" data file contains a list of arrays for each node in the circuit. These arrays contain voltage and current data for each instance in the associated time array. Once the data was imported, the component transient data was synced with the time array. Then the data from the desired time span was used to calculate efficiency. A for-loop was used to capture only the desired data after the switch transient had subsided. This was key to ensuring and accurate calculation of average power, as well as using the "nanmean" function to ignore any data points that were corrupted. The length of the time span could be adjusted in the previously mentioned setup portion of the code. For the efficiency simulations the averaging started at $0.9s$, or $0.4s$ after the turn-on transient occurred. The specific calculation processes used were covered in detail in previous chapter, Analysis and Design, Chapter: 3.4.4.5.

4.3 Thermal

Determining the thermal characteristic relationship of each switch model required the application of Equation 3.10 from Chapter 3.4.4.5 Analysis and Design. This was completed for a worst-case design, which required the maximum junction temperature, the maximum ambient temperature, the maximum operating voltage, and the maximum on-state current. As previously mentioned in the subsection 4.1 Solid-State Switch Simulation Format, not all simulated models have the same manufacturer stated junction temperature limit and drain current limit.

Therefore, to ensure a fair comparison between solid-state switches, all models will be simulated twice, just as with the switch efficiency simulations. First being at the highest junction temperatures and drain currents and again at the more limited testing parameters. The maximum testing parameters were $175°C$ for the junction temperature, $25°C$ for the ambient temperature, $1kV$ for the operating voltage, and $33\frac{1}{3}A$ for the on-state current.

The following models have a more limited junction temperature maximum of $150°C$: Device 2, Device 8, and Device 9. As well as, the following models have a more limiting on-state continuous current limit: Device 4, Device 6, Device 9, Device 10, and Device 11. Due to these limitations a more restrictive set of testing parameters were prescribed. These included a junction temperature of $150°C$ and a on-state current of $29.412A$ with all other parameters unchanged.

The simulated power dissipation of each of the solid-state switch models was determined by using Equation 3.11 from Chapter 3.4.4.5 Analysis and Design. The results from this calculation can be seen in Table 4.13. The on-resistance, ($R_{(ON)}$), of each model was surveyed from the manufacturer provided datasheets or estimated in the case of IGBTs. The maximum on-state current value, also considered the maximum drain current (I_D), or the maximum collector current (I_C), used in this calculation was derived from a simple Ohm's law calculation, Equation 1.2. The maximum input voltage, ($1kV$), was divided by the minimum load resistance, (30Ω).

Designation	$\mathbf{R_{(ON)}}$ $(m\Omega)$	$\mathbf{I_{D/C}}$ (A)	$\mathbf{P_{diss}}$ (W)
Device 1	60	33.333	66.667
Device 2	80	33.333	88.889
Device 3	80	33.333	88.889
Device 4	80	33.333	88.889
Device 5	70	33.333	78.333
Device 6	80	33.333	88.889
Device 7	70	33.333	77.778
Device 8	80	33.333	88.889
Device 9	75	33.333	83.333
Device 10	91.7	33.333	101.889
Device 11	79.2	33.333	88.000

Table 4.13: Solid-State Switch Dissipated Power

As previously mentioned, some of the chosen solid-state switches will be simulated beyond the manufacturer's recommended junction temperature limits and drain current limits. To ensure a fair comparison a separate table of dissipated power results was generated at the lower operating limits, Table 4.14. Again, the maximum on-state current value used in this calculation was derived from a simple Ohm's law calculation, Equation 1.2. The maximum input voltage, $(1kV)$, was divided by the minimum load resistance, (34Ω). The results from the more limited testing parameters will be used generate a separate less limiting thermal resistance and will be included in the decision-making process. This will highlight any drastic changes in operation at higher operating temperatures or higher operating currents and will allow for a more informed component decision.

Designation	$\mathbf{R_{(ON)}}$ $(m\Omega)$	$\mathbf{I_{D/C}}$ (A)	$\mathbf{P_{diss}}$ (W)
Device 1	60	29.412	51.9031
Device 2	80	29.412	69.2042
Device 3	80	29.412	69.2042
Device 4	80	29.412	69.2042
Device 5	70	29.412	60.9862
Device 6	80	29.412	69.2042
Device 7	70	29.412	60.5536
Device 8	80	29.412	69.2042
Device 9	75	29.412	64.8789
Device 10	80	29.412	79.3253
Device 11	75	29.412	68.5121

Table 4.14: Solid-State Switch Dissipated Power with Limited Parameters

Calculating the thermal resistance from the component case to the attached heat sink entails applying Equation 3.12. Which in turn requires calculating the surface area of each package type as well as gathering information from manufacturers of TIM. The TO-247 and SOT-227 package type dimensions used for the surface area calculations were surveyed from the manufacturer datasheets of the Device 9 and Device 3 respectively [90] [88]. The readily available TIM differs between package types, therefore so does the thermal impedance used in the calculation of thermal resistance. For all models using the SOT-227 package type the simulated TIM will be Wakefield-vette's "ulTIMiFlux Dielectric Phase Change Thermal Material", with a thermal impedance of $0.107°C - in^2/W$ at 100 PSI [93]. For all models using the TO-247 package type the simulated TIM will be Ohmite's "Thermal Interface Material for Heatsinkable Devices", with a thermal impedance of $0.18K - cm^2/W$ at 101 PSI [94].

Package	**Length**[1] (mm)	**Width**[1] (mm)	**Surface Area** (cm^2)	**Thermal Impedance** ($°C - cm^2/W$)	$\mathbf{R_{\theta,cs}}$ ($°C/W$)
SOT-227	38.1	25.3	97.35	0.6903[2]	0.007091
TO-247	20.95	15.94	33.39	0.18[3]	0.005391

Note 1: values are the average of the provided minimum and maximum
Note 2: value converted from $°C - in^2/W$
Note 3: value converted from $K - cm^2/W$

Table 4.15: Thermal Resistance of Case to Heat Sink

Deriving the thermal resistance from the heat sink to ambient ($R_{\theta,sa}$), required applying Equation 3.10. This application was completed using MATLAB's system of equation solve function. The system of equations were setup with constant attributes between each model being maximum on-state current, junction temperature, ambient temperature, and thermal resistance from case to heat sink. The variable attributes for each solid-state switch model being the power dissipated, drain to source resistance, and the thermal resistance from junction to case.

4.4 Simulation Failures

Though careful attention was paid to the simulation solve options, these alone were not able to ensure the timely function of every model over the entire operating range. For instance, the ON Semiconductor: NVHL080N120SC1 in a bidirectional circuit configuration failed to complete a full data set within 24 hours.

An investigation into decreasing the simulation duration was completed. This led to increasing the gate voltage to the maximum limit. As previously stated, this artificially adjusts the efficiency more favorably. Therefore, this model was dropped from consideration, until a more elegant solution can be found.

4.5 Energy Dissipation Circuit Format

A selection pool of devices was generated from readily available TVS diodes at electronic component distributors. The down selection criteria developed previously in Chapter 3.4.4.5 Section 3.4.3 as displayed in Table 4.16. This criteria will be applied to this substantial list as depicted in Figure 3.18. The result of this down selection process will be a manageable list of devices for comparison.

	$\mathbf{I_{PP,MIN}}$ (A)	$\mathbf{n_{EDC}}$	$\mathbf{V_{clamp,MIN}}$ (V)	$\mathbf{V_{clamp,MAX}}$ (V)	**SPICE**
Criteria	$\geq$400	$\leq$25	$\leq$1440	$\leq$1560	YES

Table 4.16: TVS Diode Selection Criteria Values

The comparison of each EDC will have two main sections: those attributes that require simulation and those that do not. Of the comparisons requiring simulation a detailed explanation will be covered in how the data was collected and processed. Additionally, any failures of EDCs that occurred in the comparison process will be reviewed in subsection 4.5.3 Simulation Failures.

4.5.1 Device Selection

The TVS diode selection pool originated with over 1440 different available devices. After applying the minimum peak pulse current limit, $I_{PP,MIN}$, the list of devices was reduced to 1187. The next selection criteria to be applied was the limit of devices in series, n_{EDC}. This resulted in a pool of 271 devices. None of the remaining devices meet the more limiting allowable clamping voltage, $V_{clamp,MIN}$. Instead, the less restrictive maximum, $V_{clamp,MAX}$, was applied resulting in 18 devices. Of these devices only 11 had an available SPICE model provided by the manufacturer. Finally, a duplicate model review was completed resulting in a comparison pool of 8 devices. The complete list of TVS diode models to be simulated is shown below in Table 4.17 [95] [96] [97] [98] [99].

Designation	**Model**	**Manufacturer**	$\mathbf{I_{PP}}$ (A)	$\mathbf{n_{EDC}}$	$\mathbf{V_{CL}}$ (V)
TVS 1	AK10-170C	Littelfuse Inc	10000	6	1560
TVS 2	AK10-380C	Littelfuse Inc	10000	3	1560
TVS 3	AK10-530C	Littelfuse Inc	10000	2	1500
TVS 4	AK3-380C	Littelfuse Inc	3000	3	1560
TVS 5	AK6-170C	Littelfuse Inc	6000	6	1560
TVS 6	AK6-380C-Y	Littelfuse Inc	6000	3	1560
TVS 7	PTVS10-380C-TH	Bourns Inc.	10000	3	1560
TVS 8	PTVS3-380C-TH	Bourns Inc.	3000	3	1560

Table 4.17: Energy Dissipation Circuit Ratings and Specifications

4.5.2 Device Comparison

Most of the EDC comparisons can be completed using the manufacturer provided data sheet information as listed below in Table 4.18. These comparisons include the total cost of the EDC, cost per power capacity of individual devices, and the activation range of the EDC. The results of these comparisons are detailed in the next chapter, Results Chapter 5.2.4. However, quantifying the clamping performance of the EDC as a whole will require simulating the devices in an example circuit.

Designation	V_{SO} (V)	$V_{BR,MIN}$ (V)	$V_{BR,MAX}$ (V)	V_{CL} (V)	I_{PP} (A)	P_{PP} (kW)	$Price_{unit}$ ($)
TVS 1	170	180	220	260	10000	2600	57.75
TVS 2	380	401	443	520	10000	5200	100.75
TVS 3	530	560	619	750	10000	7500	142.00
TVS 4	380	401	443	520	3000	1560	50.12
TVS 5	170	180	220	260	6000	1560	47.06
TVS 6	380	401	443	520	6000	3120	77.60
TVS 7	380	401	442	520	10000	5200	93.26
TVS 8	380	401	443	520	3000	1560	46.41

Table 4.18: TVS Diode Device Ratings and Specifications

4.5.2.1 EDC Simulation

Initially, each EDC was installed into a SSCB unidirectional circuit operating at highest range of current demand for 0.5 seconds of a 1.5 second transient analysis. As detailed in Figure 4.4, solid-state switch Device 9 was used as the main conduction switch with an inductive load of $100\mu H$ installed in series with the resistive load of 30Ω. The inductive transient produced after the switch opened allowed observation of the voltage clamping performance as well as the energy absorption abilities.

However, both Bourns Inc. TVS diode models failed to clamp the resultant voltage transient at a level that would prevent the body-diode of the switch from becoming reversed biased. In an attempt to prevent this type of failure, all of the simulations were re-run with each EDC installed into a SSCB bidirectional circuit, as displayed in Figure 4.5. The transient analysis timing remained the same as well as the switch component, inductive load, and resistive load. A detailed description of the failure that occurred will be covered in the following subsection 4.5.3 Simulation Failures.

4.5.2.1.1 Energy Absorption Calculation

Accomplishing the calculation of energy absorption within in LTspice was required. This is due to the nonlinear arbitrarily assigned time steps during simulation. The simulation function type allows for a max time step to be applied to the solver, but not a fixed time step. If a fixed time step was maintained throughout the simulation, then

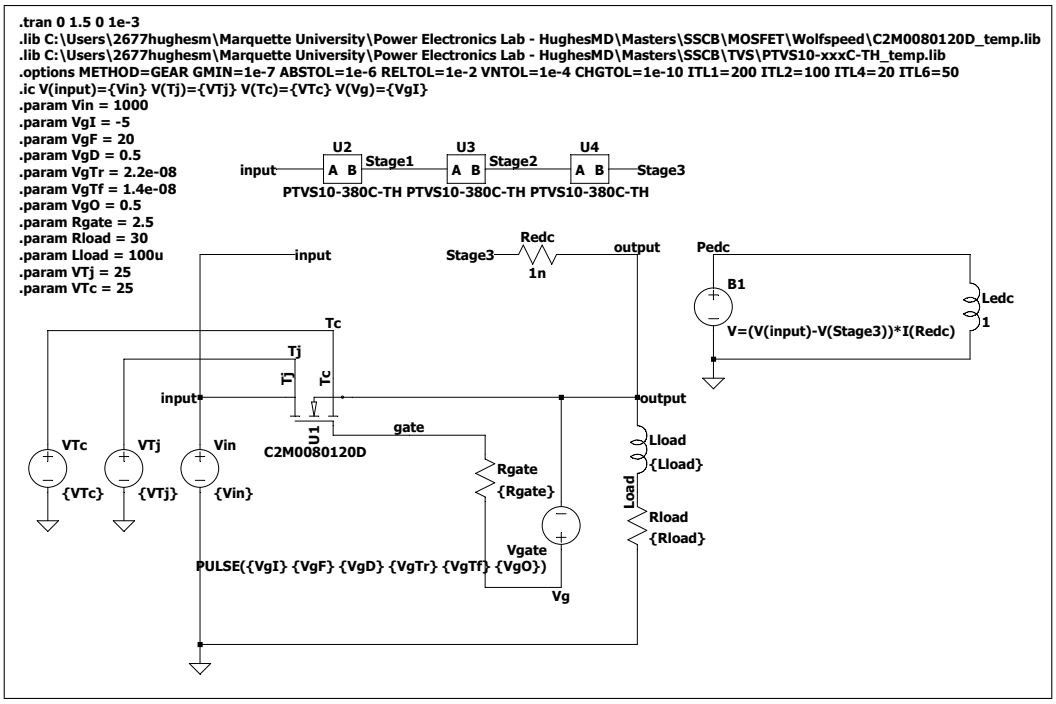


Figure 4.4: TVS Diode Unidirectional Test Circuit Example

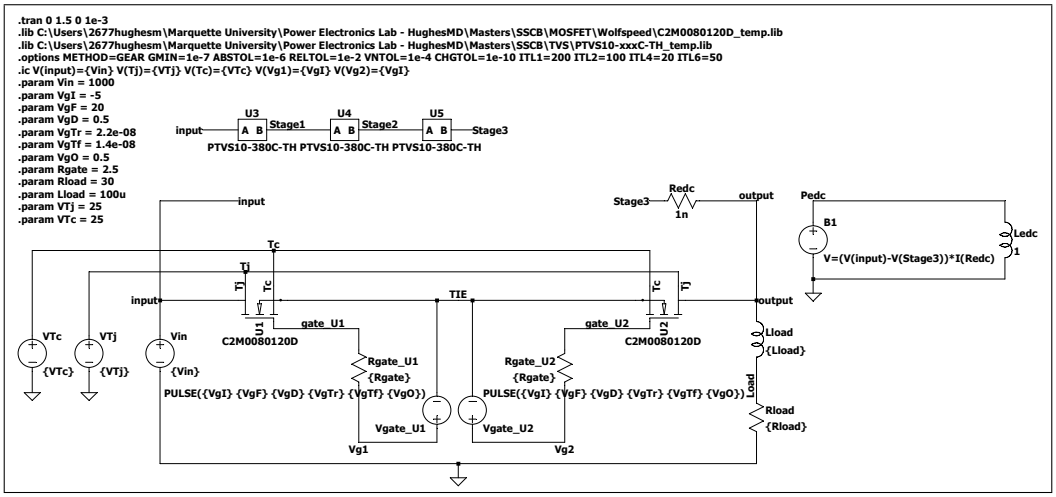


Figure 4.5: TVS Diode Bidirectional Test Circuit Example

the data could have been exported to MATLAB; just as with the efficiency calculations completed previously.

Completing the energy absorption calculation within LTspice required the use of a behavioral voltage source, also known as a "B-source". A behavioral voltage source is a function controlled voltage source, with the input being any math function. Using this type of source in conjunction with the knowledge of how to calculate energy in an inductor can be used to generate an energy absorbed over time waveform. The B-source is placed in series with an inductor with a value of $1H$, this can be seen in Figure 4.6.

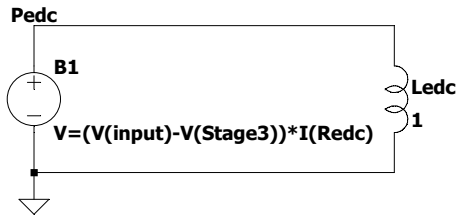


Figure 4.6: TVS Diode Test Circuit Example Behavioral Source

Using the power law, Equation 1.3, the input function is set to be the power observed by the EDC, (P_{edc}). This power is calculated as the difference in potential across the entire EDC, multiplied by the current passing through the devices. An intricacy of the B-sources requires each portion of the function to be from a single device and can not be from a multi-port device. For example, the current from a TVS diode would be annotated as "Ix(U4:2)", which will not be an acceptable input. Instead, a supplemental resistor, (R_{edc}), of low value needs to be installed in series with the device. Since the low value resistor is in series with the device, it shares the same current, similar to how a current sensing resistor would be used in a real world circuit.

Next, consider Equation 4.1. While knowing the power, (p), is the voltage being produced by the B-source, and that the value of the inductor, (L), is equal to one. Rearranging Equation 4.1, results in the current (i) of the circuit being equal to the integral of the power (p).

$$p = L\, i\frac{di}{dt} \tag{4.1}$$

Finally, consider Equation 4.2. While knowing the integral of the power, (p), is equal to the current, (i), of the circuit. Applying Equation 4.2 results in the work or energy of the device being equal to the current of the inductor. Viewing this current over time will result in energy absorption over time waveform.

$$w = \int_0^t p(t)\, dt \tag{4.2}$$

4.5.2.1.2 Energy Absorption Comparison

The data generated from the energy absorption calculation within LTspice was collected and imported to MATLAB in the same manner as with the switch efficiency calculations, using the function, "LTspice2Matlab". As a reminder, the imported ".raw" data file contains a list of arrays for each node in the circuit. These arrays contain voltage and current data for each instance in the associated time array.

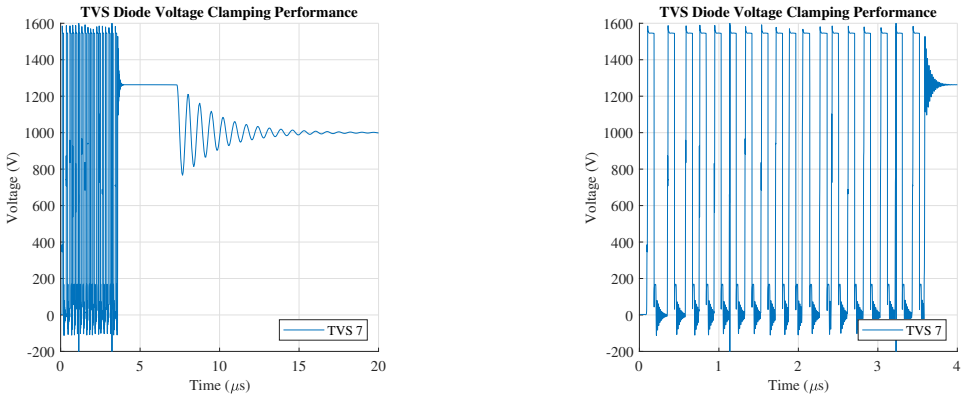
First, the time array was sampled for any occurring data errors, "NaN" values. Those columns were marked for removal in all data arrays to be used. Next, a time span was selected for observation. This time span needs to occur at the same time a potential inductive transient would, therefore at the switch opening. All other data was removed from the arrays.

The clamping voltage of the EDC is the difference between the input voltage and the voltage after the last stage of the EDC, this array is labeled as V_{edc}. As stated earlier, the power observed by the EDC is the same as the voltage being produced by the B-source, this array is labeled as P_{edc}. Again, the energy observed by the EDC is the same as the current of the inductor in series with the B-source, this array is labeled as E_{edc}. This array value would continue to increase in magnitude over the duration of the transient analysis. Since we are interested in the energy absorbed after the switch opens, the energy array is subtracted from the observed maximum value. This generates a waveform that increases over time, showcasing the energy absorbed.

4.5.3 Simulation Failures

The selection process of TVS diode models for simulation comparison was methodical, yet not foolproof. A case in point is the Bourns Inc. TVS diode models (TVS 7 and TVS 8). Here, these models met every selection criteria, but failed to perform in simulation well enough for comparison.

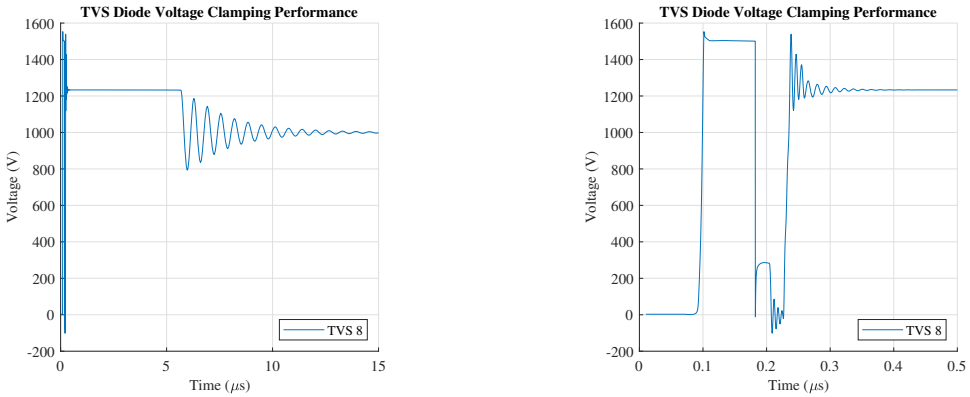
When simulated in a unidirectional circuit, both the TVS 7 and TVS 8 models failed to clamp the resultant voltage transient as calculated using Equation 3.33 from Chapter 3.4.4.5: Analysis and Design. Instead, the maximum voltage peak experienced by the TVS 7 reached $1585V$ which exceeded the maximum allowable voltage clamp limit of $1560V$. This result can be seen in Figure 4.7. Whereas in Figure 4.7b, the time scale has been reduced to highlight the successive series of voltage clamping transients.



(a) TVS Diode Full Range Clamping Transient (b) TVS Diode Zoomed Clamping Transient

Figure 4.7: TVS Diode Voltage Clamping Performance of TVS 7 in a Unidirectional SSCB Circuit Exhibiting a Maximum Peak Voltage of $1585V$

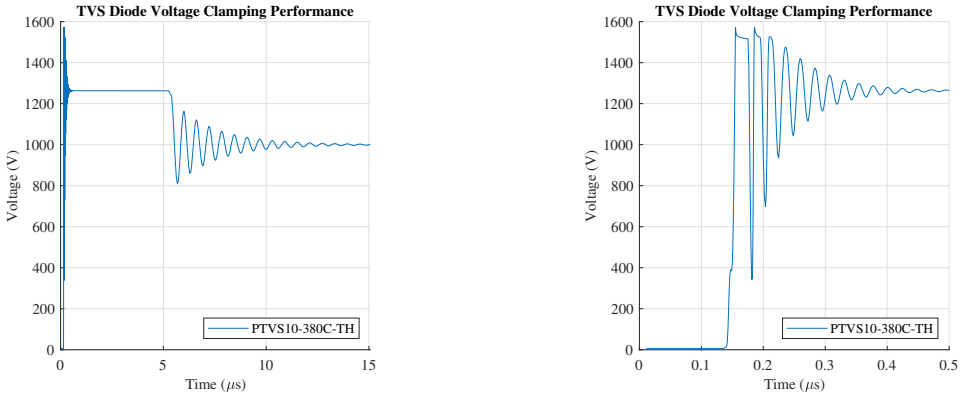
The TVS 8 experienced a maximum voltage peak of $1553V$ which did not exceed the maximum allowable voltage clamp limit of $1560V$. However, the TVS 8 did not perform as expected. This TVS diode model failed to prevent the reverse biasing of the body-diode within Device 9. This result can be seen in Figure 4.8, with a reduced time scale to highlight the successive series of voltage clamping transients in Figure 4.8b. As a reminder the maximum allowable voltage clamp limit was extended to 30% over the solid-state switch stand-off voltage limit to accommodate more TVS diode models.



(a) TVS Diode Full Range Clamping Transient (b) TVS Diode Zoomed Clamping Transient

Figure 4.8: TVS Diode Voltage Clamping Performance of TVS 8 in a Unidirectional SSCB Circuit Exhibiting a Maximum Peak Voltage of $1553V$

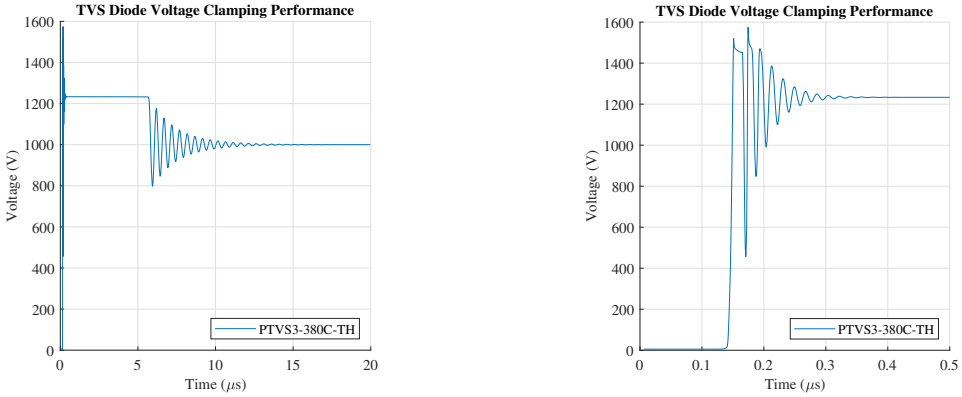
Similarly when simulated within a bidirectional SSCB circuit both TVS diode models in the selection pool from Bourns Inc. (TVS 7 and TVS 8) failed to clamp the resultant voltage transient at a level that would prevent the body-diode within Device 9 from becoming reversed biased. The TVS 7 experienced a maximum voltage peak of $1571V$ which also exceeded the maximum allowable voltage clamp limit of $1560V$. This result can be seen in Figure 4.9, with a reduced time scale to highlight the successive series of voltage clamping transients in Figure 4.9b.



(a) TVS Diode Full Range Clamping Transient (b) TVS Diode Zoomed Clamping Transient

Figure 4.9: TVS Diode Voltage Clamping Performance of TVS 7 in a Bidirectional SSCB Circuit Exhibiting a Maximum Peak Voltage of $1571V$

The trend of voltage clamping performance failures continued with TVS 8. This TVS diode model experienced a maximum voltage peak of $1575V$ which exceeded the maximum allowable voltage clamp limit of $1560V$. This result can be seen in Figure 4.10, with a reduced time scale to highlight the successive series of voltage clamping transients in Figure 4.10b.



(a) TVS Diode Full Range Clamping Transient (b) TVS Diode Zoomed Clamping Transient

Figure 4.10: TVS Diode Voltage Clamping Performance of TVS 8 in a Bidirectional SSCB Circuit Exhibiting a Maximum Peak Voltage of $1575V$

Normally, the power dissipation within the TVS diode causes a change in avalanche voltage. Eventually the avalanche voltage returns to normal after the transient. Both of the above simulated Bourns Inc. models have the same stand-off voltage limit, but different pulse peak power characteristics. Because the successive series of clamping transients was worsened in the higher pulse peak power rated device, TVS 7, I can assume this cycling is related to this power limit or by extension the pulse peak current limit.

The results from the aforementioned simulation discussion will be covered in Chapter 5.2.4: Results of this thesis. As previously mentioned, any insights reached from the results will be explained in Chapter 6.6: Conclusion and Future Work.

CHAPTER 5
RESULTS

The following sections will fully explain the results from the analytical relationships developed earlier as well as the outcomes from the simulation phase. These results will lead to a final decision on primary components to be used as well as explain the reasoning in the decision-making process.

5.1 Solid-State Switch

The simulation results of the eleven solid-state switch models are described below. The results are divided into subsections of efficiency, thermal, and costing. The efficiency results have an additional subsection due to each of the solid-state switches being simulated in two different circuit configurations.

5.1.1 Efficiency

The efficiencies will be displayed as the change in continuous drain current occurs. As a reminder, this change in drain current is due to the change in load demand. Also as a reminder, efficiency is expected to drop from the "I^2R" losses, as noted in previous chapters. This drop in efficiency will occur across all temperature ranges. An overview of all the efficiencies will be displayed showing the low end, ambient, and high end junction temperature limits. Following, will be a breakdown of each of the temperature ranges. In those breakdowns any results worth noting will be briefly discussed. Following the graphical results, a detailed review of discrete data points will be conducted.

As previously mentioned, not all of the compared solid-state switch models have the same junction temperature limit or drain current limit. To ensure consistent comparisons, all of the outcomes will be displayed and compared including those outside the recommended operating limits. After the results from the highest junction temperature ($175°C$), a review of the the results from a more limited junction temperature ($150°C$), will be completed. As well as a review of the discrete data points from the more limited load demand of 34Ω.

5.1.1.1 Unidirectional Circuit

The simulated results of the unidirectional solid-state switch circuit can be seen in Figure 5.1. In this figure, each solid-state switch model that was simulated was assigned an individual color. This color assignment will be maintained through out all of the displayed results. Also, within each assigned color, contrasting plot line styles were used to denote the different junction temperatures: -55°C, 25°C, and 175°C. Again, these contrasting plot line styles will be maintained through out all of the display results, unless otherwise noted.

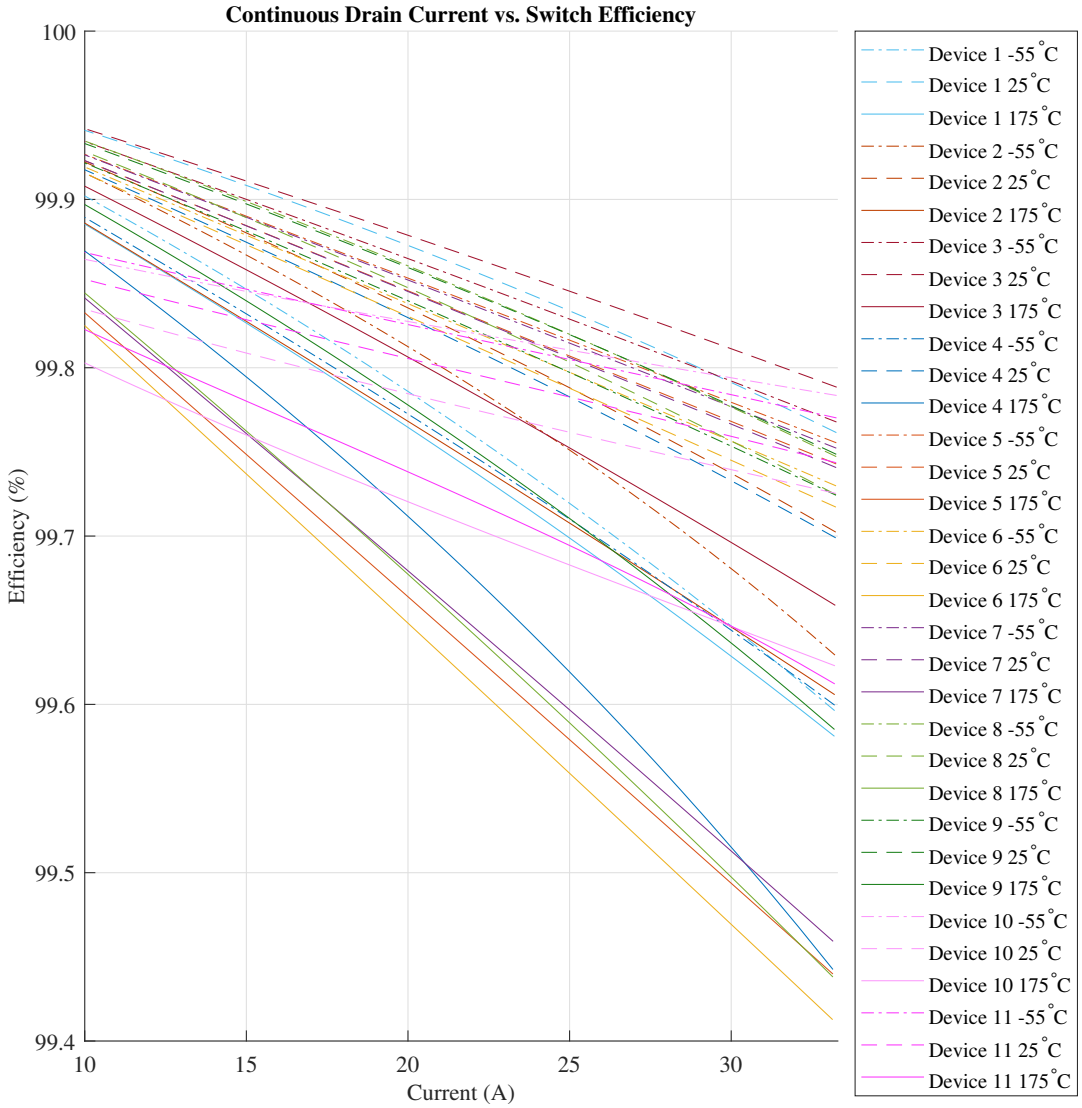


Figure 5.1: Unidirectional Switch Efficiency versus Continuous Drain Current

The simulated efficiency results at a junction temperature of -55°C can be seen in Figure 5.2. One out of the eleven solid-state switches simulated measurably out performed the rest of the selection pool at the high end of the current demand range. Device 10 began with less than -0.07% margin in efficiency under Device 8 at the low end of the current demand range, ending with a margin greater than +0.013% over Device 11 at the highest end. It is worth noting that Device 10 and Device 11 were simulated above their recommended collector current limit of 30A. Exceeding these limits may have artificially skewed the results.

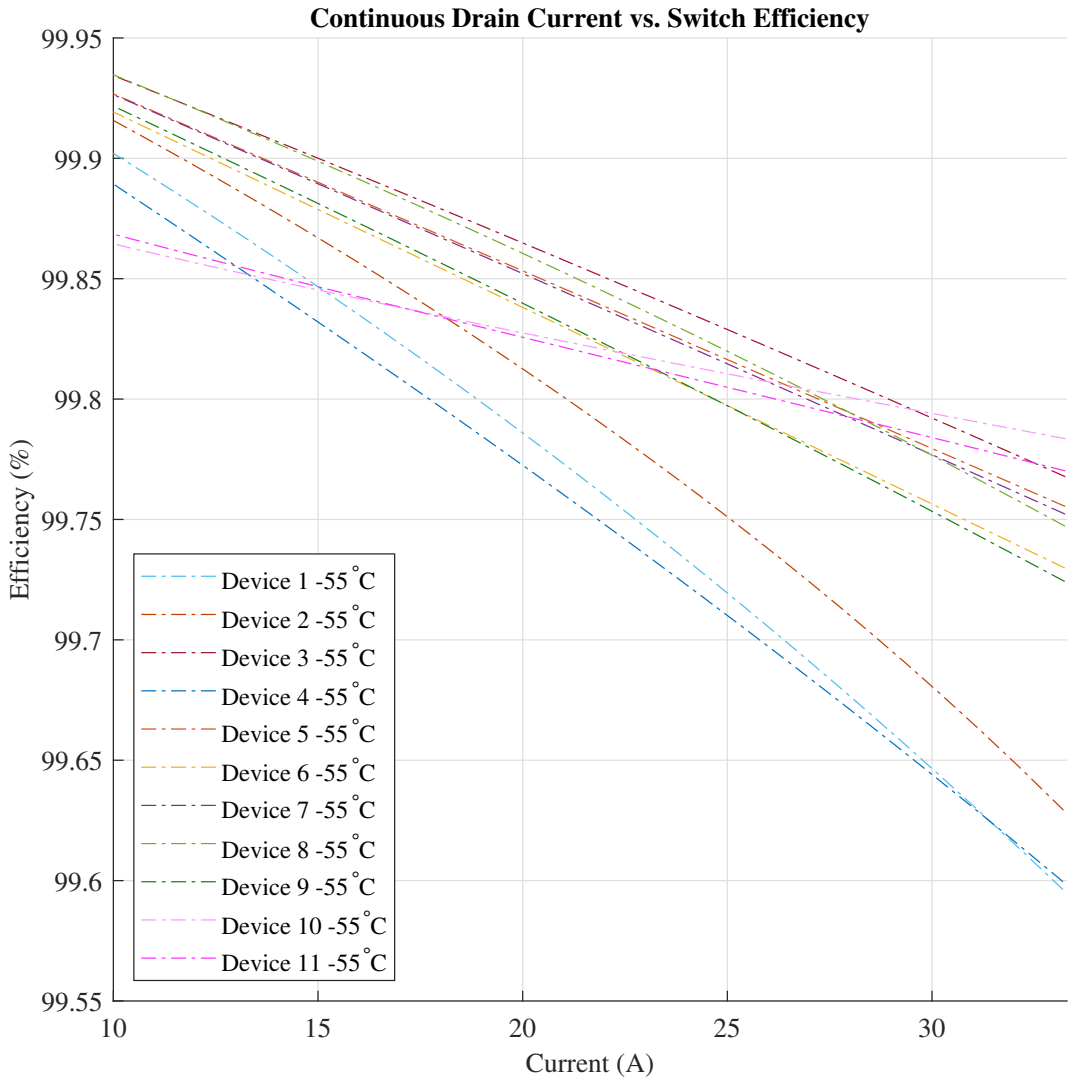


Figure 5.2: Unidirectional Switch Efficiency versus Continuous Drain Current at a Junction Temperature of -55°C

The simulated efficiency results at a junction temperature of $25°C$ can be seen in Figure 5.3. One out of the eleven solid-state switches simulated measurably out performed the rest of the selection pool at the high end of the current demand range. Device 3 began with less than a +0.001% margin in efficiency above Device 1 at the low end of the current demand range, ending with a margin greater than +0.026% at the highest end. It is worth noting that neither of these models were simulated above its recommended drain current limit.

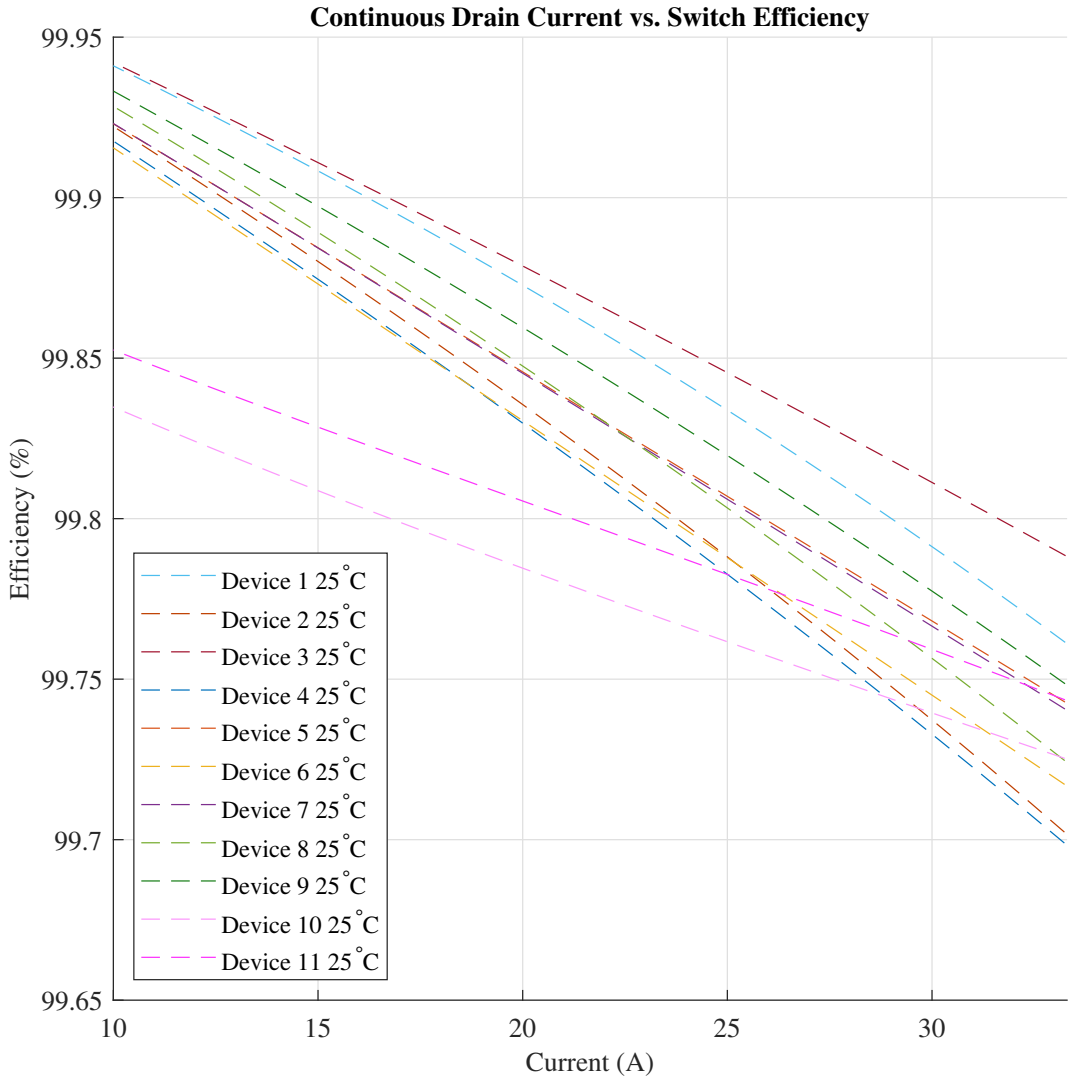


Figure 5.3: Unidirectional Switch Efficiency versus Continuous Drain Current at a Junction Temperature of $25°C$

The simulated efficiency results at a junction temperature of $175°C$ can be seen in Figure 5.4. One out of the eleven solid-state switches simulated measurably out performed the rest of the selection pool at the high end of the current demand range. Device 3 began with less than a +0.010% margin in efficiency above Device 9 at the low end of the current demand range, ending with a margin greater than +0.035% over Device 10 at the highest end. It is worth noting that Device 9 was simulated above its recommended drain current limit of $30A$ and junction temperature limit of $150°C$. Device 10 was also simulated above its recommended collector current limit of $30A$. Exceeding these limits may have artificially skewed the results.

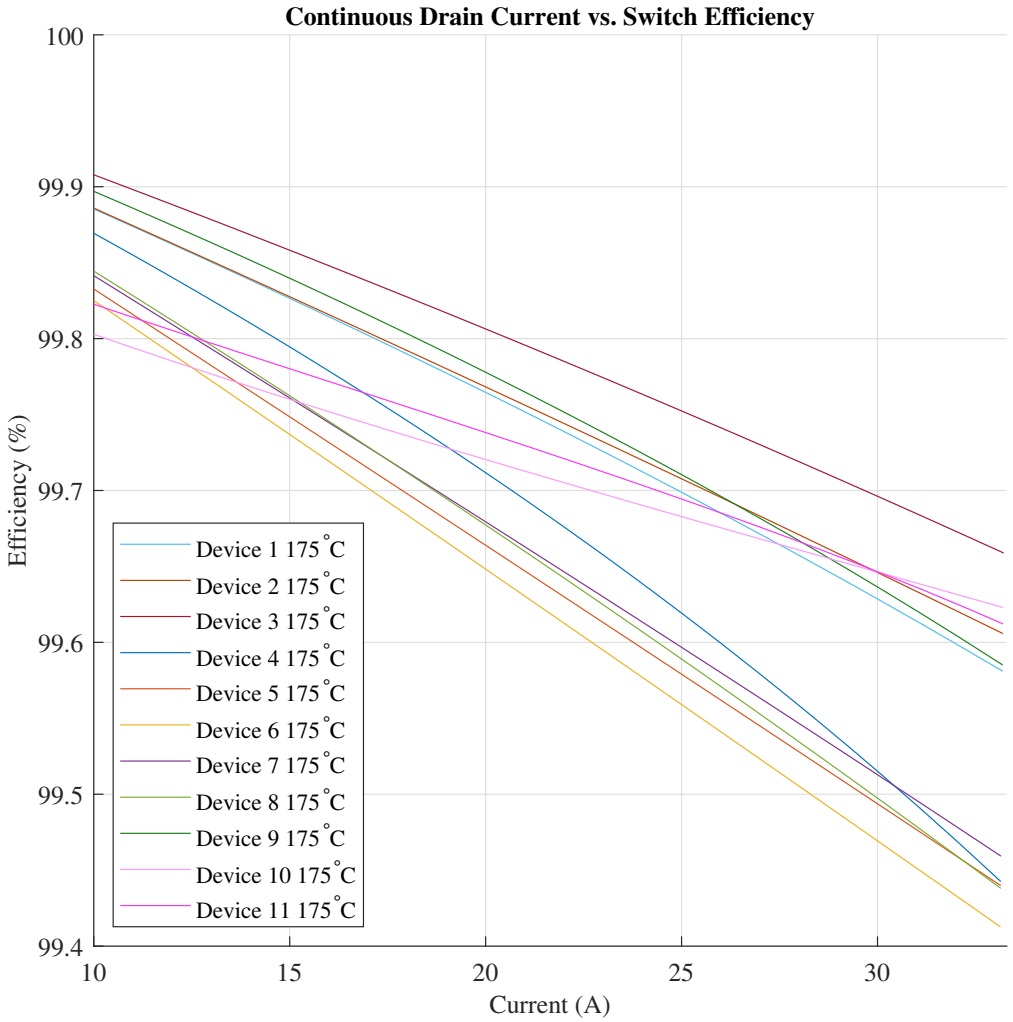


Figure 5.4: Unidirectional Switch Efficiency versus Continuous Drain Current at a Junction Temperature of $175°C$

A discrete view of each unidirectional solid-state switch efficiency can be found in Table 5.1. This table displays the low end, (100Ω), and high end, (30Ω), current demand efficiencies at each simulated junction temperatures: -55°C, 25°C, and 175°C. A gradient heat map has been applied to the discreet values table, with green indicating the highest efficiency, red indicating lowest efficiency, and yellow being the midpoint.

Designation	**-55°C**		**25°C**		**175°C**	
	100Ω	**30Ω**	**100Ω**	**30Ω**	**100Ω**	**30Ω**
Device 1	99.90216	99.59630	99.94109	99.76145	99.88547	99.58104
Device 2	99.91580	99.62930	99.92224	99.70235	99.88606	99.60577
Device 3	99.93442	99.76776	99.94207	99.78845	99.90794	99.65884
Device 4	99.88943	99.59955	99.91768	99.69895	99.86953	99.44251
Device 5	99.92680	99.75542	99.92305	99.74286	99.83294	99.43981
Device 6	99.91933	99.72982	99.91558	99.71720	99.82535	99.41270
Device 7	99.92652	99.75222	99.92314	99.74069	99.84168	99.45922
Device 8	99.93472	99.74721	99.92848	99.72463	99.84468	99.43804
Device 9	99.92174	99.72424	99.93323	99.74858	99.89707	99.58511
Device 10	99.86449	99.78353	99.83486	99.72532	99.80304	99.62293
Device 11	99.86856	99.77018	99.85259	99.74356	99.82284	99.61215

Table 5.1: Discrete Unidirectional Circuit Efficiencies

The efficiency of each circuit lowered as continuous on-state current increased, regardless of temperature range. This tracks with the earlier assertion that the line losses would be present and measurable. This result reiterates the importance of understanding the ohmic characteristics of each of the switches used.

The ambient temperature range of 25°C experienced the highest overall efficiencies. These results assert that the solid-state switches operate more efficiently at lower temperatures, but do not require extraordinarily low temperatures. This result effects the thermal operating necessities and therefore, may impact overall cost as well. A more detailed review of this result will be completed in the subsection 5.1.2 Thermal and 5.1.3 Costing.

These discrete values have been used to make broad assertions but, also can be used to generate more granulated conclusions. In particular, they will be used as inputs to a decision matrix. This decision matrix will be realized in a following chapter, Conclusions and Future Work, Chapter 6.6.

The simulated results of the unidirectional solid-state switch circuit at a more limited operating temperature and load limit can be seen in Figure 5.5. In this figure, each solid-state switch model that was simulated was assigned an individual color. This color assignment will be maintained through out all of the displayed results. Also, within each assigned color, contrasting plot line styles were used to denote the different junction temperatures: -55°C, 25°C, and 150°C. Again, these contrasting plot line styles will be maintained through out all of the display results, unless otherwise noted.

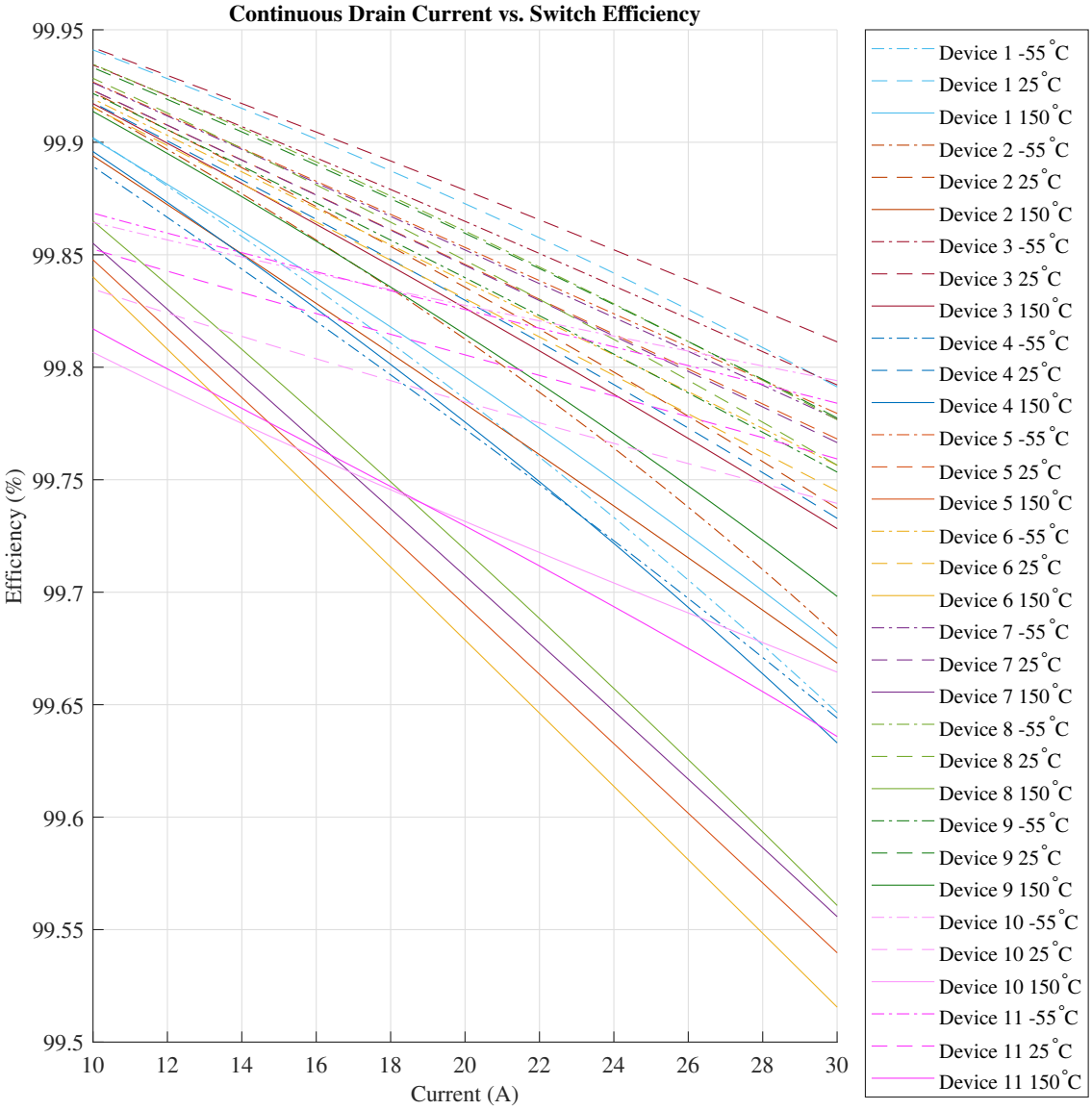


Figure 5.5: Unidirectional Switch Efficiency versus Continuous Drain Current at Limited Operating Parameters

The simulated efficiency results at a junction temperature of $150°C$ can be seen in Figure 5.6. One out of the eleven solid-state switches simulated measurably out performed the rest of the selection pool at the high end of the current demand range. Device 3 began with less than a +0.003% margin in efficiency above Device 9 at the low end of the current demand range, ending with a margin greater than +0.028% at the highest end. These results are limited to a continuous drain current of less than $30A$ and may provide a more accurate outlook of performance conversely to the previous higher temperature and load simulations.

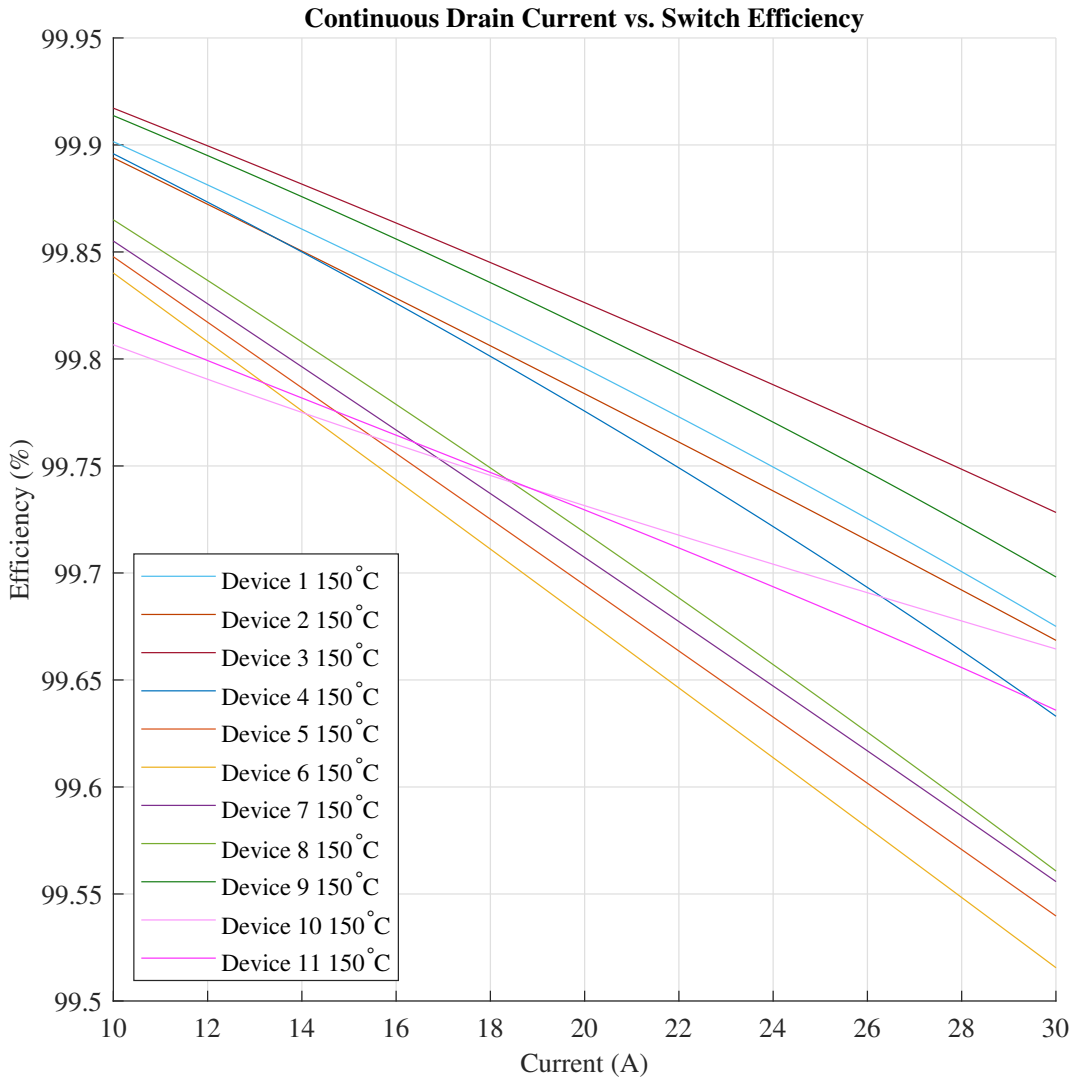


Figure 5.6: Unidirectional Switch Efficiency versus Continuous Drain Current at a Junction Temperature of $150°C$

A discrete view of each unidirectional solid-state switch efficiency operating at the more limited testing parameters can be found in Table 5.2. This table displays the low end, (100Ω), and more limited high end, (34Ω), current demand efficiencies at each simulated junction temperature: -55°C, 25°C, and 150°C. A gradient heat map has been applied to the discreet values table, with green indicating the highest efficiency, red indicating lowest efficiency, and yellow being the midpoint.

Designation	**-55°C**		**25°C**		**150°C**	
	100Ω	**34Ω**	**100Ω**	**34Ω**	**100Ω**	**34Ω**
Device 1	99.90216	99.65700	99.94109	99.7970	99.90162	99.68385
Device 2	99.91580	99.69085	99.92224	99.74425	99.89406	99.67658
Device 3	99.93442	99.79693	99.94207	99.81572	99.91724	99.73504
Device 4	99.88943	99.65344	99.91768	99.73966	99.89598	99.64382
Device 5	99.92680	99.78429	99.92305	99.77321	99.84797	99.55091
Device 6	99.91933	99.76181	99.91558	99.75068	99.84041	99.52750
Device 7	99.92652	99.78182	99.92314	99.77170	99.85530	99.56678
Device 8	99.93472	99.78246	99.92848	99.76280	99.86518	99.57250
Device 9	99.92174	99.75925	99.93323	99.78309	99.91378	99.70668
Device 10	99.86449	99.79615	99.83486	99.74238	99.80678	99.66896
Device 11	99.86856	99.78673	99.85259	99.76238	99.81721	99.64288

Table 5.2: Discrete Unidirectional Circuit Efficiencies at Limited Parameters

Just as with the previous results, the efficiency of each circuit lowered as continuous on-state current raised. Also a repeated result, the ambient temperature range of 25°C experienced the highest overall efficiencies.

In regards to the best performing device in a unidirectional circuit, the result was unchanged when simulated at a more limited junction temperature and load current demand. Each device experienced improved performance, most notably were the devices that were previously simulated past their design limits. These devices became more competitive overall and experienced less variance in efficiency across the range of load demand.

These discrete values along with the previous data points from the less limited testing parameters of 175°C junction temperature and 30Ω full load, will be used as inputs to a decision matrix as detailed in Conclusions and Future Work, Chapter 6.6.

5.1.1.2 Bidirectional Circuit

The simulated results of the bidirectional solid-state switch circuit can be seen in Figure 5.7. In this figure, each solid-state switch model that was simulated was assigned an individual color. This color assignment will be maintained through out all of the displayed results. Also, within each assigned color, contrasting plot line styles were used to denote the different junction temperatures: -55°C, 25°C, and 175°C. Again, these contrasting plot line styles will be maintained through out all of the display results, unless otherwise noted.

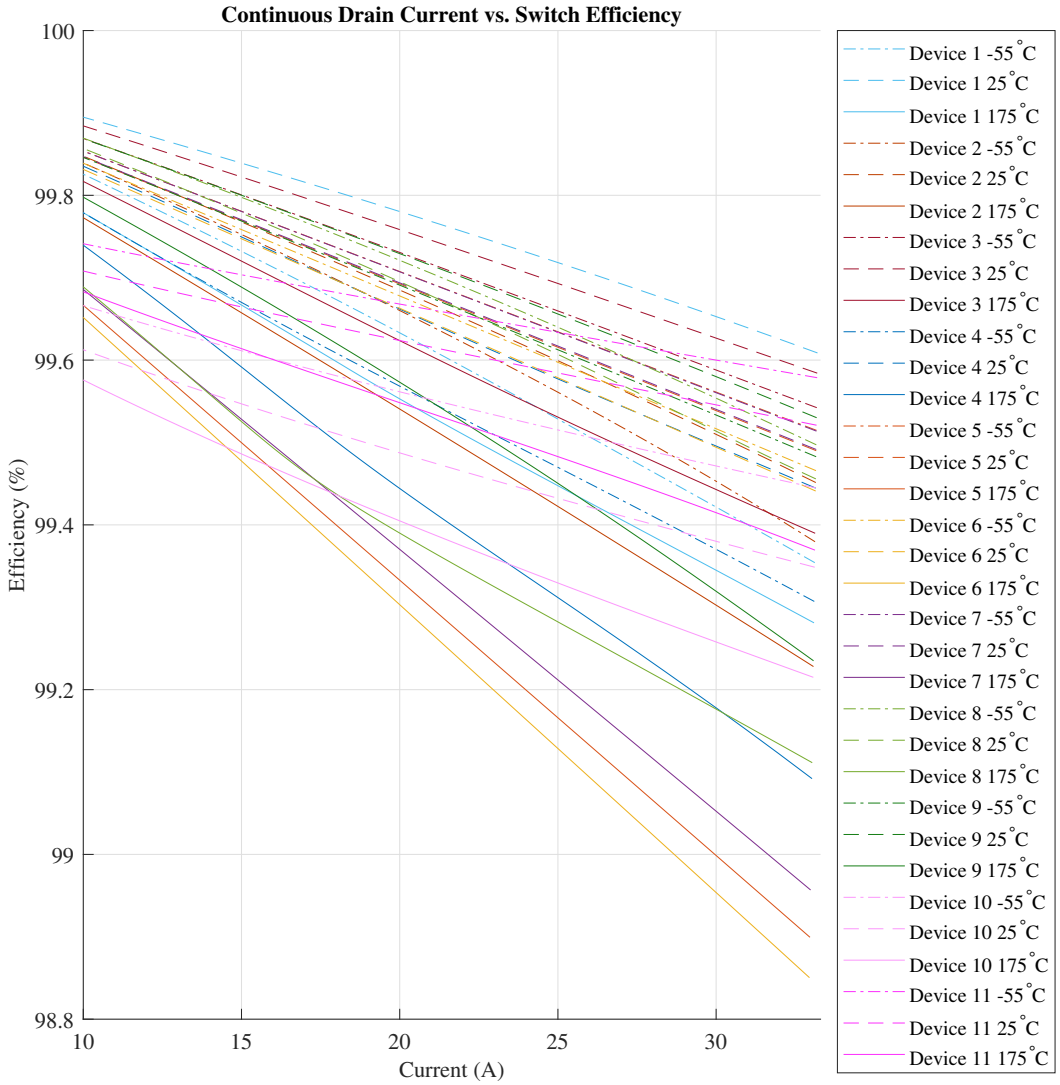


Figure 5.7: Bidirectional Switch Efficiency versus Continuous Drain Current

The simulated efficiency results at a junction temperature of -55°C can be seen in Figure 5.8. One out of the eleven solid-state switches simulated measurably out performed the rest of the selection pool at the high end of the current demand range. Device 11 began with less than -0.13% margin in efficiency under Device 8 at the low end of the current demand range, ending with a margin greater than +0.036% over Device 3 at the highest end. It is worth noting that Device 10 was simulated above it's recommended collector current limit of 30A. Exceeding these limits may have artificially skewed the results.

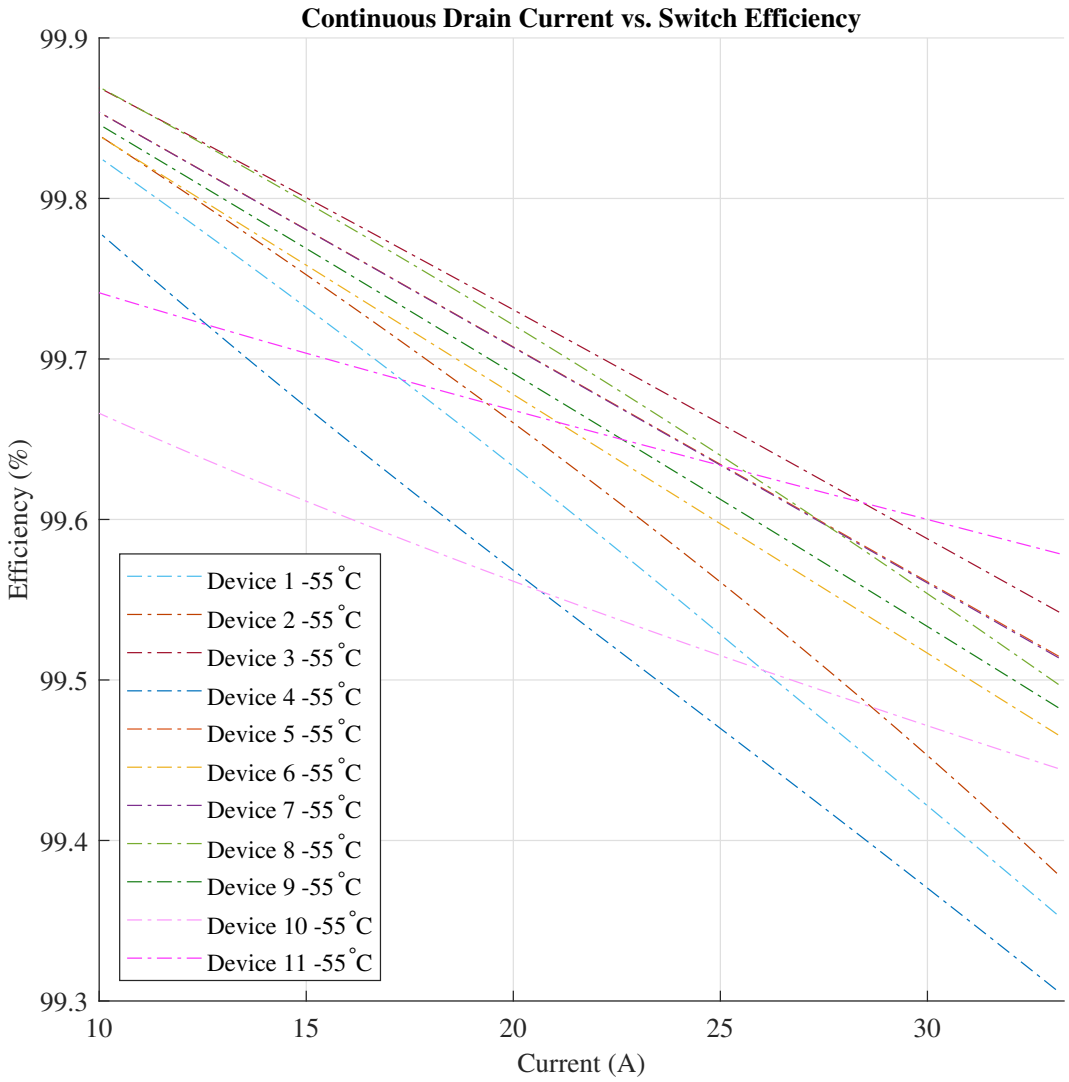


Figure 5.8: Bidirectional Switch Efficiency versus Continuous Drain Current at a Junction Temperature of -55°C

The simulated efficiency results at a junction temperature of $25°C$ can be seen in Figure 5.9. One out of the eleven solid-state switches simulated measurably out performed the rest of the selection pool at the high end of the current demand range. Device 1 began with less than a +0.011% margin in efficiency above Device 3 at the low end of the current demand range, ending with a margin greater than +0.024% at the highest end. It is worth noting that neither of these models were simulated above its recommended drain current limit.

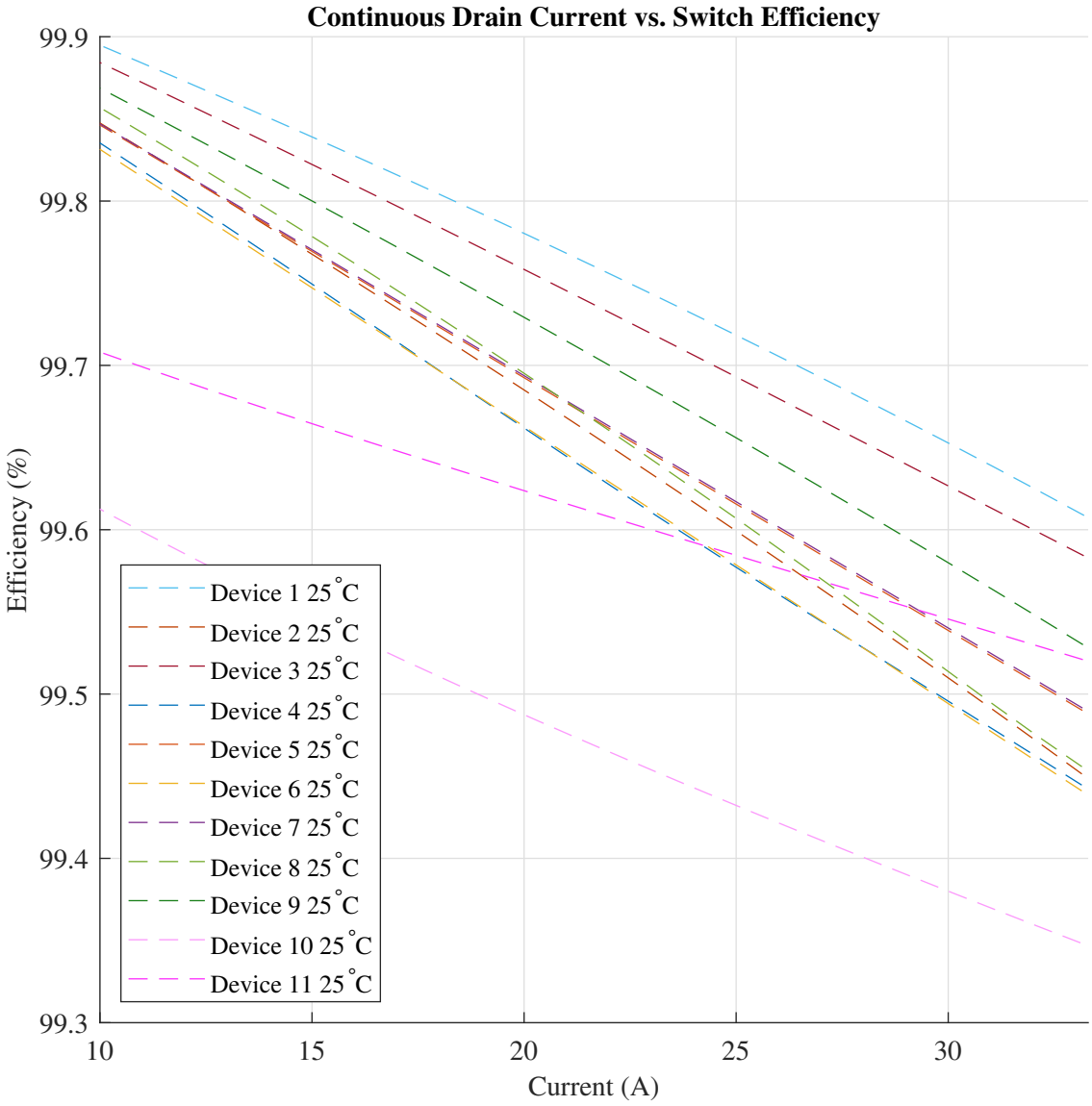


Figure 5.9: Bidirectional Switch Efficiency versus Continuous Drain Current at a Junction Temperature of $25°C$

The simulated efficiency results at a junction temperature of $175°C$ can be seen in Figure 5.10. One out of the eleven solid-state switches simulated measurably out performed the rest of the selection pool at the high end of the current demand range. Device 3 began with less than a +0.019% margin in efficiency above Device 9 at the low end of the current demand range, ending with a margin greater than +0.020% at the highest end. It is worth noting that Device 9 model was simulated above its recommended drain current limit of $30A$ and junction temperature limit of $150°C$. As well as Device 11 being simulated above its recommended collector current limit of $30A$. Exceeding these limits may have artificially skewed the results.

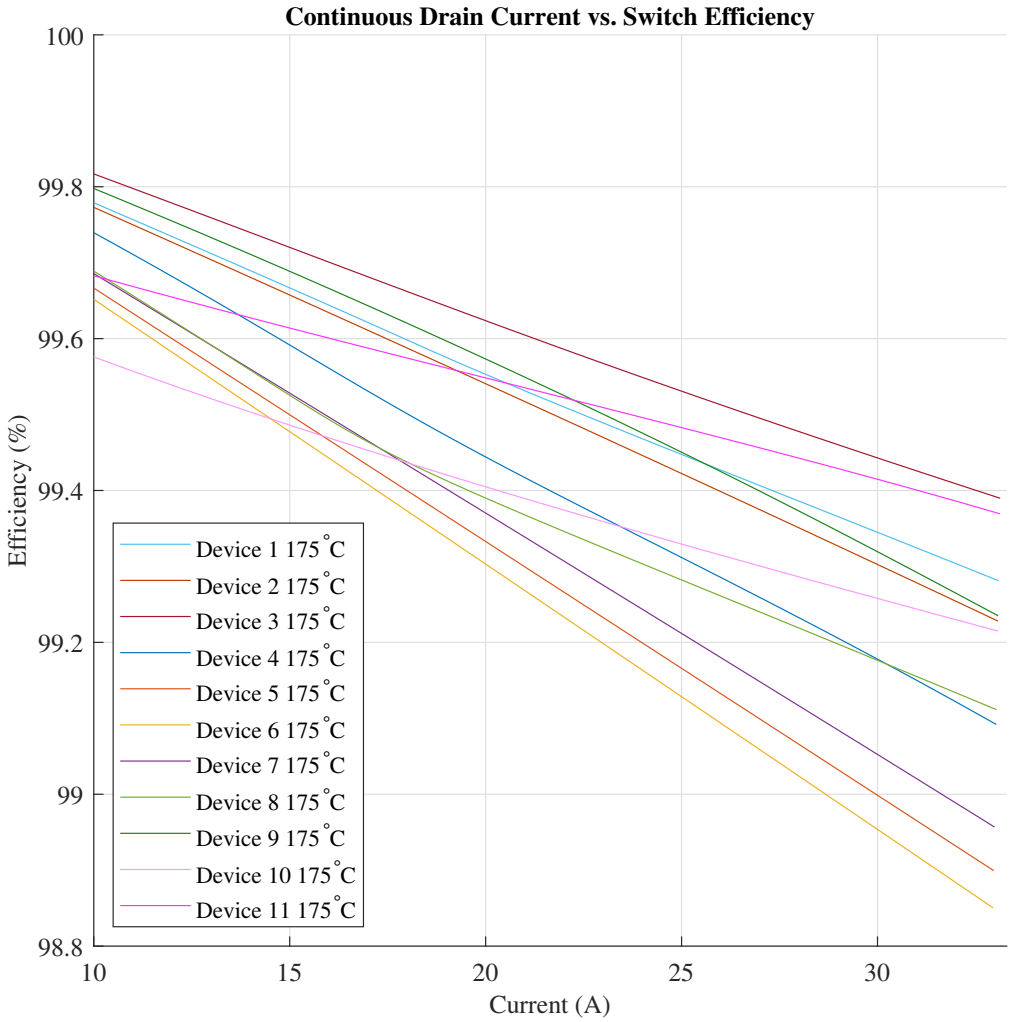


Figure 5.10: Bidirectional Switch Efficiency versus Continuous Drain Current at a Junction Temperature of $175°C$

A discrete view of each bidirectional solid-state switch efficiency can be found in Table 5.3. This table displays the low end, (100Ω), and high end, (30Ω), current demand efficiencies at each simulated junction temperatures: -55°C, 25°C, and 175°C. A gradient heat map has been applied to the discreet values table, with green indicating the highest efficiency, red indicating lowest efficiency, and yellow being the midpoint.

Designation	**-55°C**		**25°C**		**175°C**	
	100Ω	**30Ω**	**100Ω**	**30Ω**	**100Ω**	**30Ω**
Device 1	99.82595	99.35412	99.89497	99.60868	99.77935	99.28107
Device 2	99.83952	99.37952	99.84773	99.45136	99.77335	99.22802
Device 3	99.86912	99.54226	99.88440	99.58417	99.81720	99.38978
Device 4	99.77930	99.30707	99.83562	99.44460	99.74019	99.09197
Device 5	99.85396	99.51475	99.84650	99.49002	99.66765	98.89940
Device 6	99.83922	99.46580	99.83176	99.44108	99.65295	98.85040
Device 7	99.85388	99.51376	99.84718	99.49150	99.68669	98.95682
Device 8	99.86955	99.49718	99.85708	99.45578	99.68989	99.11135
Device 9	99.84636	99.48272	99.86887	99.53005	99.79813	99.23502
Device 10	99.66651	99.44491	99.61307	99.34860	99.57666	99.21495
Device 11	99.74145	99.57863	99.70838	99.52095	99.68330	99.36934

Table 5.3: Discrete Bidirectional Circuit Efficiencies

A reiteration of the unidirectional circuit results, the efficiency lowered as on-state current raised; though the result is more drastic due to the added on-state resistance of the second solid-state switch. Also, this result can be readily seen across the board regardless of temperature range, with the ambient temperature range of 25°C experiencing the highest overall efficiencies.

The best performing device overall, Device 3, did not change from the unidirectional circuit; though it was out performed in the ambient temperature range by the Device 1. Why the results did not remain consistent between circuit configurations is unknown. A possibility is the modeling of the on-state resistance of each device. The listed on-state resistances as seen in Table 4.1 are singular and do not reflect how a device would operate over a change in junction temperature and current demand.

These discrete values along with the previous data points from the unidirectional circuit, will be used as inputs to a decision matrix as detailed in Conclusions and Future Work, Chapter 6.6.

The simulated results of the bidirectional solid-state switch circuit at a more limited operating temperature and load limit can be seen in Figure 5.11. In this figure, each solid-state switch model that was simulated was assigned an individual color. This color assignment will be maintained through out all of the displayed results. Also, within each assigned color, contrasting plot line styles were used to denote the different junction temperatures: -55°C, 25°C, and 150°C. Again, these contrasting plot line styles will be maintained through out all of the display results, unless otherwise noted.

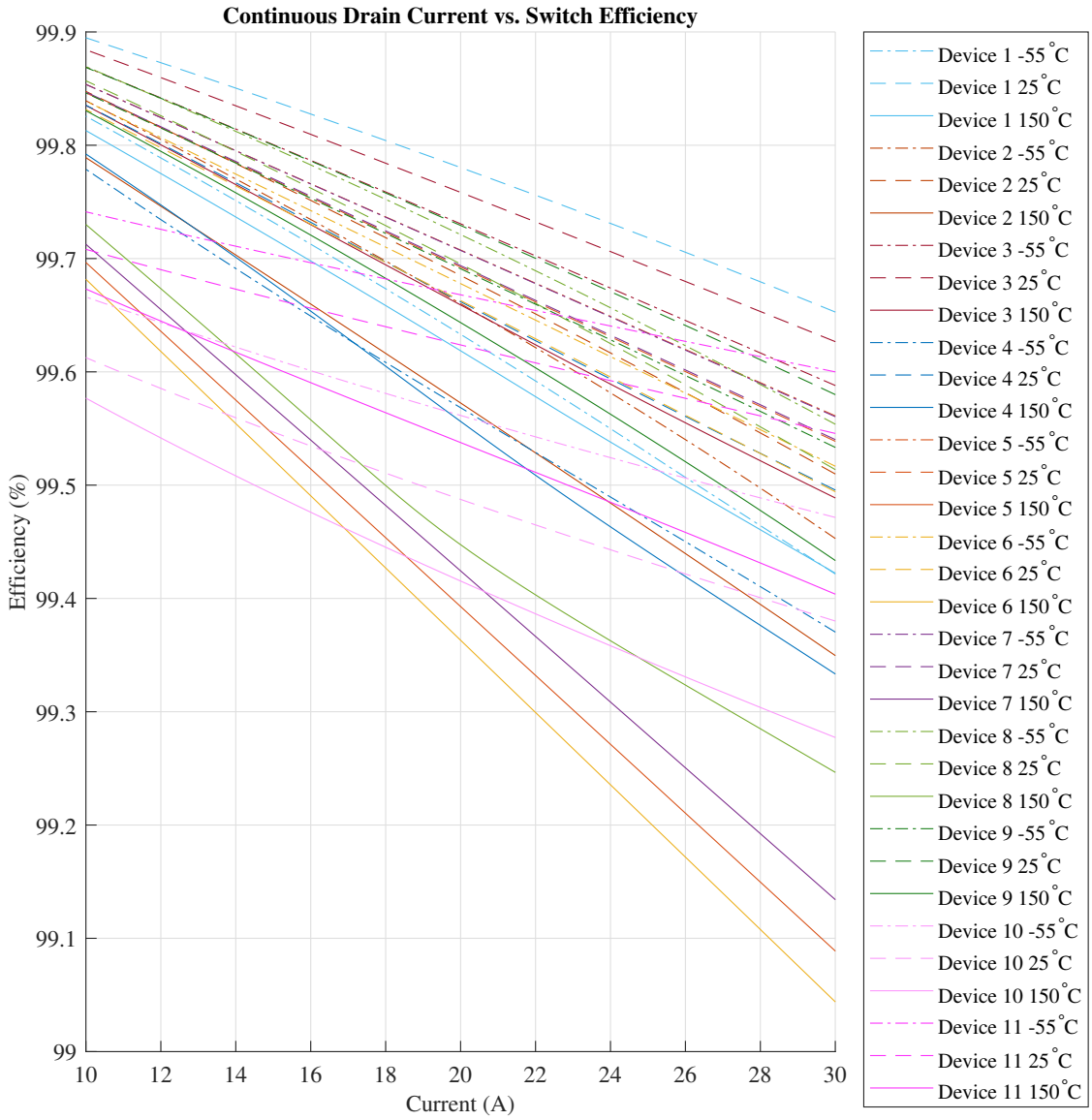


Figure 5.11: Bidirectional Switch Efficiency versus Continuous Drain Current at Limited Operating Parameters

The simulated efficiency results at a junction temperature of $150°C$ can be seen in Figure 5.12. One out of the eleven solid-state switches simulated measurably out performed the rest of the selection pool at the high end of the current demand range. Device 3 began with less than a +0.005% margin in efficiency above Device 9 at the low end of the current demand range, ending with a margin greater than +0.050% at the highest end. These results are limited to a continuous drain current of less than $30A$ and may provide a more accurate outlook of performance conversely to the previous higher temperature and load simulations.

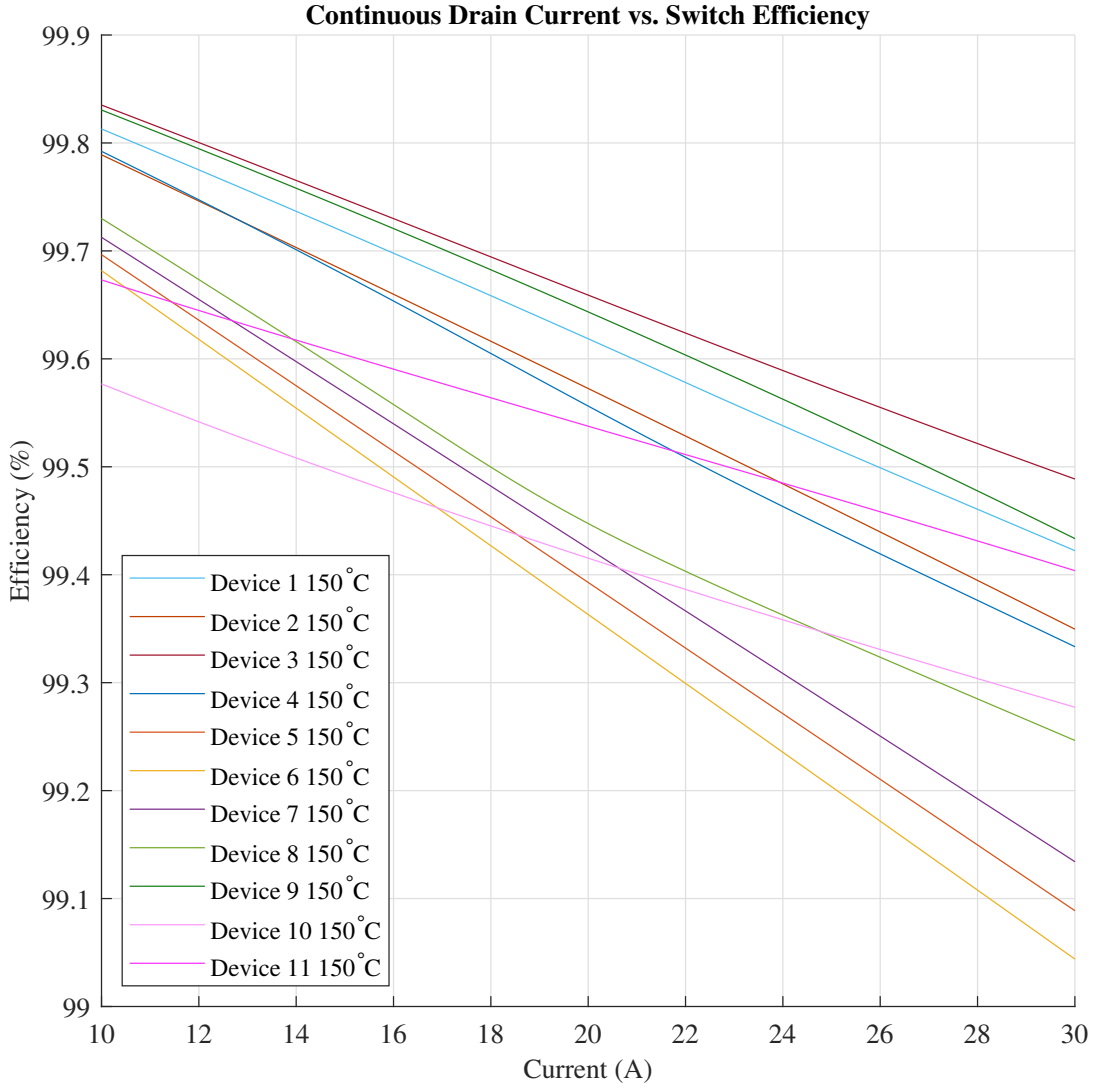


Figure 5.12: Bidirectional Switch Efficiency versus Continuous Drain Current at a Junction Temperature of $150°C$

A discrete view of each bidirectional solid-state switch efficiency can be found in Table 5.4. This table displays the low end, (100Ω), and high end, (34Ω), current demand efficiencies at each simulated junction temperatures: -55°C, 25°C, and 150°C. A gradient heat map has been applied to the discreet values table, with green indicating the highest efficiency, red indicating lowest efficiency, and yellow being the midpoint.

Designation	**-55°C**		**25°C**		**150°C**	
	100Ω	**34Ω**	**100Ω**	**34Ω**	**100Ω**	**34Ω**
Device 1	99.82595	99.43771	99.89497	99.66203	99.81330	99.43675
Device 2	99.83952	99.46968	99.84773	99.52312	99.78947	99.36708
Device 3	99.86912	99.59808	99.88440	99.63591	99.83538	99.50066
Device 4	99.77930	99.38579	99.83562	99.50753	99.79270	99.35008
Device 5	99.85396	99.57161	99.84650	99.54978	99.69742	99.11460
Device 6	99.83922	99.52841	99.83176	99.50657	99.68271	99.07144
Device 7	99.85388	99.57090	99.84718	99.55126	99.71342	99.15850
Device 8	99.86955	99.56640	99.85708	99.52747	99.73076	99.26199
Device 9	99.84636	99.54480	99.86887	99.59085	99.83079	99.45020
Device 10	99.66651	99.47784	99.61307	99.38790	99.57754	99.28779
Device 11	99.74145	99.60470	99.70838	99.55121	99.67358	99.41433

Table 5.4: Discrete Bidirectional Circuit Efficiencies at Limited Parameters

Just as with the previous results, the efficiency of each circuit lowered as continuous on-state current raised. Also a repeated result, the ambient temperature range of 25°C experienced the highest overall efficiencies.

As expected, simulating the devices at a reduced load current demand and junction temperature increased the operating performance across the board. This result was similarly reached by other research as explained in Chapter 2.5: Review of Literature, Section 2.4.1.2. Though, it is worth noting the devices that were previously simulated pass their listed design limits experienced larger gains in performance, but none large enough to be significant. In regards to the best performing device in a bidirectional circuit, the result was unchanged from the previous results and no new insights were gained.

These discrete values along with the previous data points from the less limited testing parameters of 175°C junction temperature and 30Ω full load, will be used as inputs to a decision matrix as detailed in Conclusions and Future Work, Chapter 6.6.

5.1.2 Thermal

The thermal attributes used to determine the heat sinking requirements of each solid-state switch were input variables into Equation 3.10 from Chapter 3.4.4.5 Analysis and Design. The results from applying Equation 3.10 derived the maximum allowable thermal resistance from the attached heat sink to ambient air. As previously noted, a larger thermal resistance value is more desirable due to more available heat sinking options. These thermal resistances from the highest junction temperature ($175°C$) and on-state current ($30\frac{1}{3}A$), can be seen in Table 5.5.

Designation	$\mathbf{P_{diss}}$ (W)	$\mathbf{R_{\theta,jc}}$[1] $(°C/W)$	$\mathbf{R_{\theta,cs}}$ $(°C/W)$	$\mathbf{T_j}$ $(°C)$	$\mathbf{T_a}$ $(°C)$	$\mathbf{R_{\theta,sa}}$ $(°C/W)$
Device 1	66.6667	1[2]	0.005391	175	25	1.2446
Device 2	88.8889	0.70	0.005391	175	25	0.9821
Device 3	88.8889	0.75	0.007091	175	25	0.9304
Device 4	88.8889	0.91	0.005391	175	25	0.7721
Device 5	78.3334	0.59	0.005391	175	25	1.3195
Device 6	88.8889	0.59	0.005391	175	25	1.0921
Device 7	77.7778	0.59	0.005391	175	25	1.3332
Device 8	88.8889	0.65	0.005391	175	25	1.0321
Device 9	83.3334	1.10	0.005391	175	25	0.6946
Device 10	101.889	0.74[2]	0.005391	175	25	0.7268
Device 11	88.0000	0.63[2]	0.005391	175	25	1.0692

Note 1: $R_{\theta,jc}$ values are at maximum
Note 2: manufacturer data sheet provided values in K/W

Table 5.5: Solid-State Switch Heat Sink Requirements

Three of the simulated solid-state switches were outliers having the least limiting thermal resistance from sink to ambient, ($R_{\theta,sa}$). Device 5 and Device 7 as well as the Device 1 all had thermal resistances exceeding $1.2°C/W$.

Each of these outliers had the least amount of power being dissipated, due to their low on-state resistance ($R_{DS(ON)}$). One would also expect the solid-state switch with the highest efficiency to have a low power dissipation value. The power dissipation of Device 9 was in the middle of the pack, but the manufacturer provided thermal resistance from the junction to the case, ($R_{\theta,jc}$), was higher than any other and in some cases double. In the end this lead to the most restrictive thermal resistance value.

As previously mentioned, some of the chosen solid-state switches were simulated beyond the manufacturer's recommended junction temperature and drain current limits. To ensure a fair comparison a separate table of results was generated at the lower junction temperature ($150°C$) and on-state current ($29\frac{2}{5}A$), as can be seen in Table 5.6.

Designation	$\mathbf{P_{diss}}$ (W)	$\mathbf{R_{\theta,jc}}$[1] $(°C/W)$	$\mathbf{R_{\theta,cs}}$ $(°C/W)$	$\mathbf{T_j}$ $(°C)$	$\mathbf{T_a}$ $(°C)$	$\mathbf{R_{\theta,sa}}$ $(°C/W)$
Device 1	51.9031	1[2]	0.005391	150	25	1.4029
Device 2	69.2042	0.7	0.005391	150	25	1.1009
Device 3	69.2042	0.75	0.007091	150	25	1.0492
Device 4	69.2042	0.91	0.005391	150	25	0.8909
Device 5	60.9862	0.59	0.005391	150	25	1.4543
Device 6	69.2042	0.59	0.005391	150	25	1.2109
Device 7	60.5536	0.59	0.005391	150	25	1.4689
Device 8	69.2042	0.65	0.005391	150	25	1.1509
Device 9	64.8789	1.1	0.005391	150	25	0.8213
Device 10	79.3253	0.74[2]	0.005391	150	25	0.8304
Device 11	68.5121	0.63[2]	0.005391	150	25	1.1891

Note 1: $R_{\theta,jc}$ values are at maximum

Note 2: manufacturer data sheet provided values in K/W

Table 5.6: Solid-State Switch Heat Sink Requirements at Limited Parameters

The same three simulated solid-state switches were outliers having the least limiting thermal resistance from sink to ambient, ($R_{\theta,sa}$). Device 5 and Device 7 as well as the Device 1 all had thermal resistances exceeding $1.4°C/W$.

Again, each of these outliers had the least amount of power being dissipated. Also again, Device 9 which was a highly efficient switch model in any configuration at any temperature or at any current demand, had a power dissipation value in the middle of the pack. The relatively large manufacturer provided thermal resistance from the junction to the case, ($R_{\theta,jc}$), did not change. Therefore, again lead to this model having the most restrictive thermal resistance value out of the selection pool.

It is worth noting that Device 9 is recommended to be operated at the lower junction temperature and drain current. This switch model is the only one in the selection pool to be limited in both areas of temperature and current. Though, Table 5.6 displays these limited parameters and the resultant thermal resistance was still the most restrictive.

Additionally of note, none of the top three performing solid-state switch models exceeded the manufacturer recommended junction temperature or drain current limits in either data set. As well as noting that the solid-state switch models that did exceed their recommended limits were not consistently at the bottom nor performed more than relatively better under favorable conditions. In particular, Device 9 did perform better in the second data set, but so too did all the other solid-state switch models. This relative change in performance can be seen in Figure 5.13

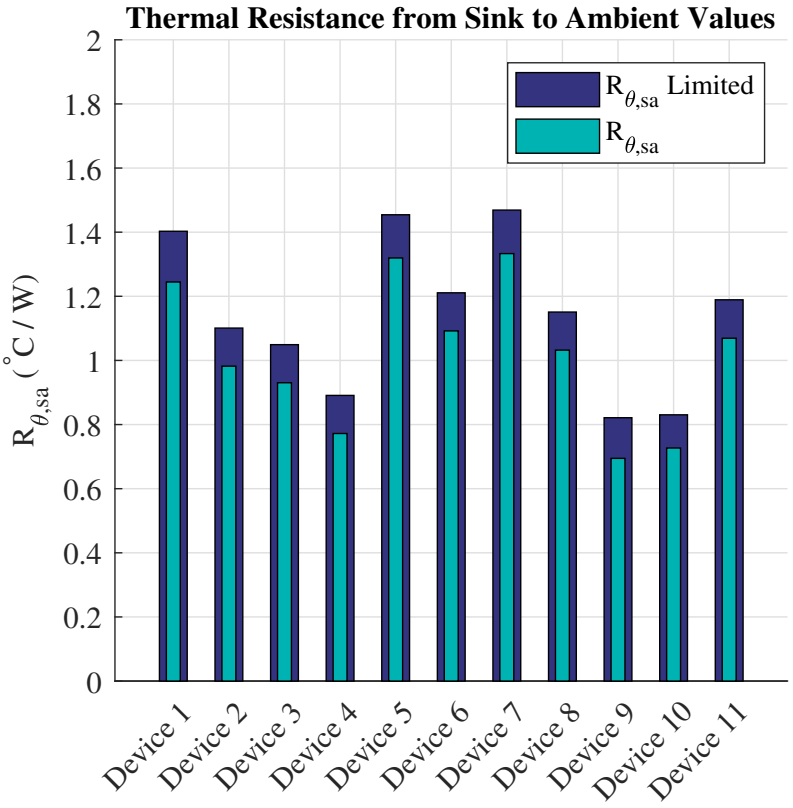


Figure 5.13: Relative Change in Thermal Resistance from Sink to Ambient ($R_{\theta,sa}$)

The change in performance between the simulation conditions can be seen to be relative. As in each of the solid-state switch models performed the same in both simulated conditions, relative to each other. This result concludes that the specific simulation conditions are negligible. What is more important is that each device being compared is simulated under the same parameters.

5.1.3 Costing

Determining the price per current density, ($Price_{ampere}$, $\$/ampere$), required the application of Equation 3.17 from Chapter 3.4.4.5 Analysis and Design. The results from applying Equation 3.17 can be seen in Table 5.7. Applying the equation appropriately required limiting the on-state current, ($I_{D/C}$, $ampere$), parameters to typical values at ambient temperature. As well as using the price per unit, ($Price_{unit}$, $\$/unit$), cost of a single unit with no price break provided by the component distributor for bulk purchasing.

Designation	$\mathbf{I_{D/C}}$ (A)	**Price** ($\$/unit$)	$\mathbf{Price_{ampere}}$ ($\$/ampere$)
Device 1	36	15.45	0.4292
Device 2	39	15.12	0.3877
Device 3	37	25.13	0.6792
Device 4	31	18.23	0.5880
Device 5	33.5	12.60	0.3761
Device 6	33	12.60	0.3818
Device 7	34.5	14.39	0.4171
Device 8	36	16.67	0.4631
Device 9	30	24.05	0.8017
Device 10	30	5.70	0.1900
Device 11	30	5.11	0.1703

Table 5.7: Solid-State Switch Cost per Unit Ampere

As seen in Table 5.7, two of the eleven simulated solid-state switches were outliers having the lowest price per ampere. Device 10 and Device 11 both had a price per ampere below $\$0.20/A$.

The IGBT models, Device 10 and Device 11, had the lowest unit price of the selection pool and the lowest on-state current capability. Of the FET models, Device 5 and Device 6, had lowest unit price and a mid-range on-state current capability.

The low unit price of the IGBTs led to a low price per current and could signify a better overall cost for a project using multiple solid-state switches. Though cost is only one aspect of a project and other attributes should be taken into account for a more holistic comparison. Just as with the device efficiencies, these discrete values will be utilized a decision matrix.

Another view of the same comparison can be seen in Figure 5.14. Here, the unit price of each device is displayed aligned to the left axis. In conjunction with the unit price, the results from Table 5.7, price per ampere are displayed aligned to the right axis.

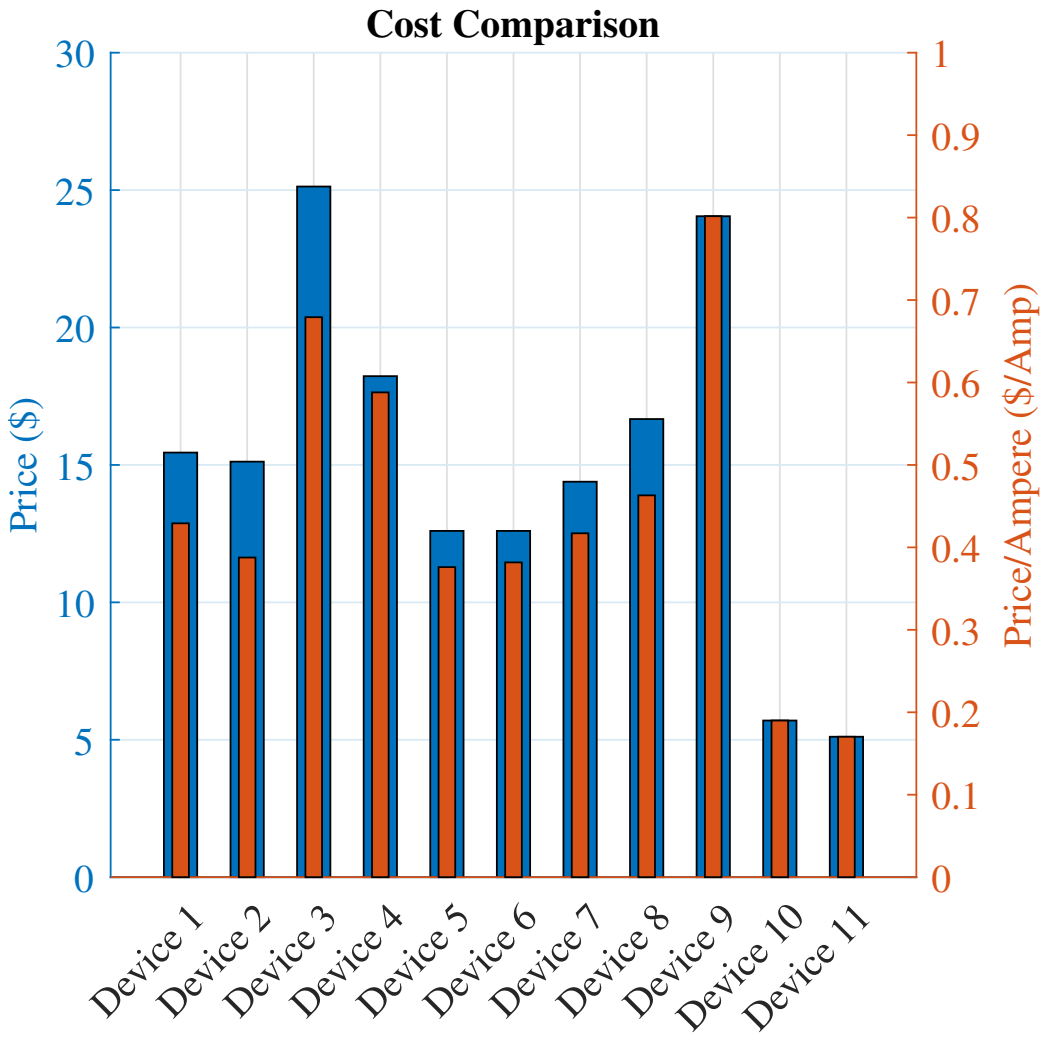


Figure 5.14: Solid-State Switch Cost Comparison

In Figure 5.14, a clustering of FET devices can be seen with a unit price near or below $15.00. These same devices also have a favorable price per ampere. Keeping in mind the competitive marketplace of device manufacturing, one could assume most devices have similar performance metrics. This assumption with the anecdotal evidence in Figure 5.14, one could assume targeting devices with a unit price near or below $15.00 would yield a better price per ampere.

5.2 Energy Dissipation Device

The analytical and simulation results of the eight TVS diode models are described below. The results are divided into subsections of costing, activation range, clamping performance, and energy absorption. These results will not be divided into different circuit configurations. As seen in Figure 3.13 and Figure 3.14 from Chapter 3.4.4.5 Analysis and Design, a single energy dissipation circuit can be used for either solid-state switch configuration.

5.2.1 Costing

Aspects of costing under consideration in terms of the energy dissipation circuit are overall cost and cost with respect to power rating. Determining the total price, ($Price_{total}$, \$), required the application of Equation 3.38. Determining the price per power rating, ($Price_{watt}$, $\$/kW$), required the application of Equation 3.39 as detailed in Chapter 3.4.4.5 Analysis and Design. The results from applying these equations can be seen in Table 5.8.

Designation	$\mathbf{n_{EDC}}$	$\mathbf{P_{PP}}$ (kW)	$\mathbf{Price_{unit}}$ (\$)	$\mathbf{Price_{total}}$ (\$)	$\mathbf{Price_{watt}}$ ($\$/kW$)
TVS 1	6	2600	57.75	346.50	0.02221
TVS 2	3	5200	100.75	302.25	0.01938
TVS 3	2	7500	142.00	284.00	0.01893
TVS 4	3	1560	50.12	150.36	0.03213
TVS 5	6	1560	47.06	282.36	0.03017
TVS 6	3	3120	77.60	232.50	0.02484
TVS 7	3	5200	93.29	279.87	0.01794
TVS 8	3	1560	46.41	139.23	0.02975

Table 5.8: Energy Dissipation Circuit Power and Price Comparison

As seen in Table 5.8, three of the eight simulated TVS diodes were outliers having the lowest price per watt. The TVS 2 and TVS 3 as well as the TVS 7 all had a price per watt below $\$0.20/kW$.

In most cases the peak pulse power of each outlier was more than double the other devices. This difference was enough to overcome the high unit prices. Though the total price of the outliers should be reviewed for a better overall solution.

Another view of the same comparison can be seen in Figure 5.15. Here, the unit price of each device is displayed aligned to the left axis as well as the total price of the energy dissipation circuit. In conjunction with the unit price and total price, the results from Table 5.8, price per watt are displayed aligned to the right axis.

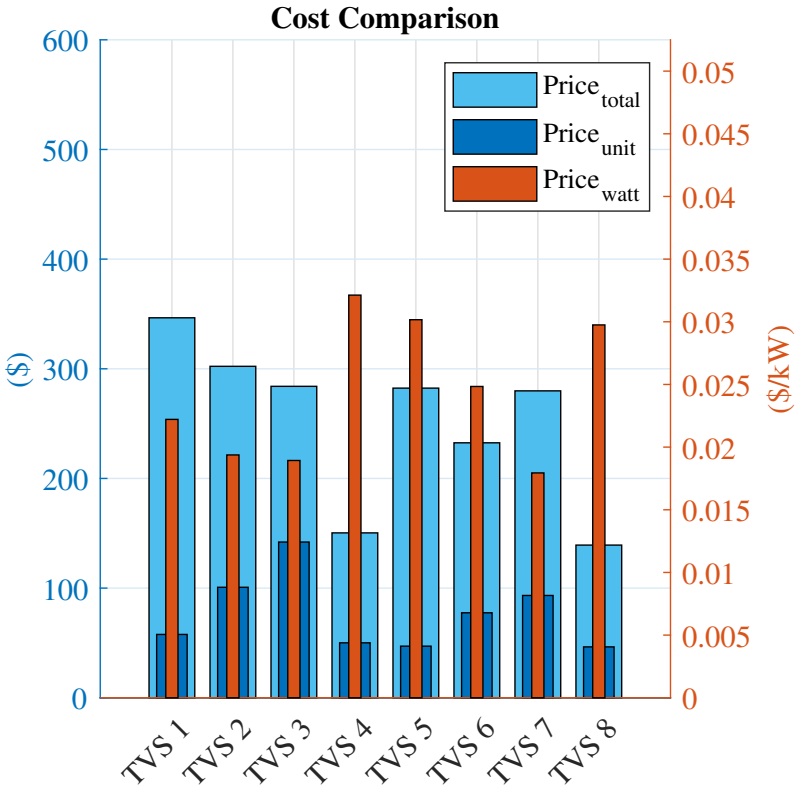


Figure 5.15: TVS Diode Cost Comparison

In Figure 5.15, the benefit of a mid-range device can be more easily seen. As can be seen with TVS 6, which is well performing in all areas, but does not top any metric. This is due to the varied number of devices required. Unlike with the solid-state switches a robust solution can only be achieved after knowing the application design parameters. Furthermore the price per unit may vary with bulk purchasing as the value used was the cost of a single unit with no price break provided by an electronic component distributor.

5.2.2 Activation Range

The voltage attributes used to determine the activation and breaking voltage ranges of each energy dissipation circuit were input variables into Equation 3.41 and Equation 3.42. The results from applying these equations from Chapter 3.4.4.5 Analysis and Design can be seen in Table 5.9. As a reminder, smaller values for each of these ranges is more desirable.

Designation	$\mathbf{n_{EDC}}$	$\mathbf{V_{SO}}$ (V)	$\mathbf{V_{BR,MIN}}$ (V)	$\mathbf{V_{BR,MAX}}$ (V)	$\mathbf{V_{CL}}$ (V)	$\mathbf{V_{active}}$ (V)	$\mathbf{V_{break}}$ (V)
TVS 1	6	1020	1080	1320	1560	540	240
TVS 2	3	1140	1203	1329	1560	420	126
TVS 3	2	1060	1120	1238	1500	440	118
TVS 4	3	1140	1203	1329	1560	420	126
TVS 5	6	1020	1080	1320	1560	540	240
TVS 6	3	1140	1203	1329	1560	420	126
TVS 7	3	1140	1203	1326	1560	420	123
TVS 8	3	1140	1203	1326	1560	420	123

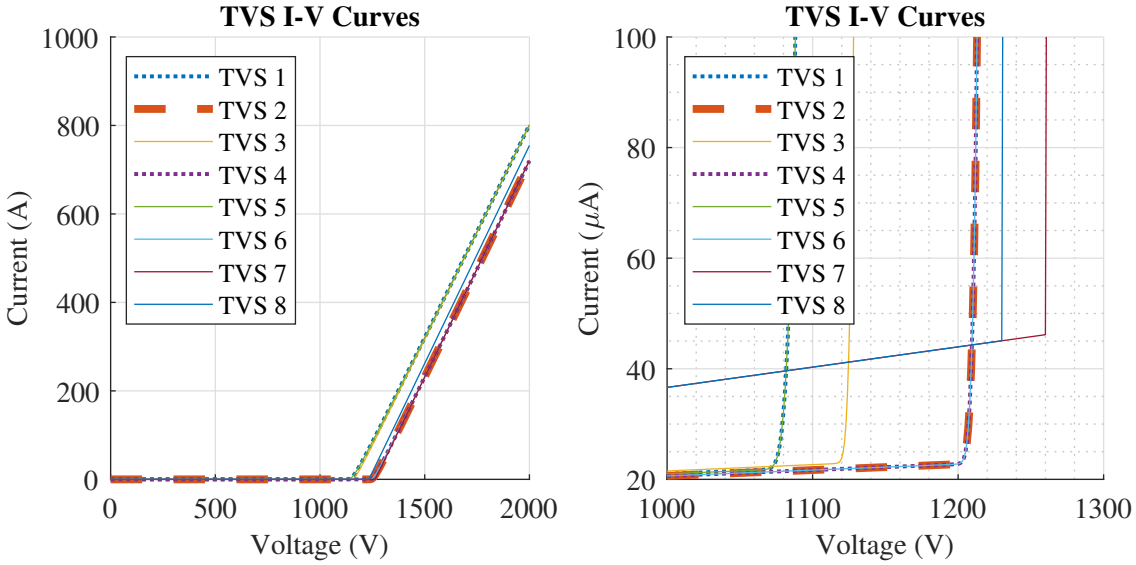
Table 5.9: Energy Dissipation Circuit Voltage Comparisons

Five of the eight simulated energy dissipation circuits had the smallest voltage activation range. TVS 2, TVS 4, TVS 6, TVS 7, and TVS 8 all had a range of $420V$. Followed by TVS 3 with a range of $440V$. Leaving TVS 1 and TVS 5 with the largest range of $540V$.

All but one of the circuits, TVS 3, simulated had the same clamping voltage of $1560V$, leaving the circuit stand-off voltage to be the delineation factor. This grouping of devices has a common individual stand-off voltage of $380V$ and by extension the same circuit stand-off voltage of $1140V$. Requiring a further review of the breaking voltage range. In this group of five devices, there were two stand outs in terms of breaking voltage range. TVS 7 and TVS 8 had a smaller breaking voltage range by $3V$ at $123V$.

It is worth noting that the simulated unidirectional circuit of TVS 7 failed to clamp before the activation of the body-diode. This in-turn required the re-simulation of the energy dissipation circuits in a bidirectional switch circuit instead, as noted in the previous chapter, Section 4.5.2.

Looking more closely at the simulated current versus voltage curves of the devices can help verify the accuracy of the data sheet provided by the manufacturer. In Figure 5.16a, the full range of the energy dissipation circuit activation is displayed for reference. In Figure 5.16b, the axes have been adjusted to highlight the knee of the curve, with the current ranging from $20\mu A$ to $100\mu A$.



(a) Energy Dissipation Circuit Full Range (b) Energy Dissipation Circuit Break-over Range

Figure 5.16: Energy Dissipation Circuit I-V Curve Comparison

In Figure 5.16b, each of the Littelfuse Inc. models, TVS 1 through TVS 6, have a knee point consistent with the minimum breaking voltage limit. The Bourns Inc. models, TVS 7 and TVS 8, have a knee at a higher voltage than listed. Specifically, TVS 7 has a knee that is consistent with the mid-point of the breaking voltage range and TVS 8 has a knee below the mid-point by approximately $30V$.

The testing condition cited on each of the manufacturer provided data sheets lists the test current at $10mA$. At the test condition current TVS 7 and TVS 8 have a voltage greater than the minimum breaking voltage listed, but were still below the mid-point of the breaking voltage range by approximately $30V$. This puts TVS 1 through TVS 6 near the same point as TVS 7 and TVS 8, but with the added benefit of starting the break-over consistently at minimum breaking voltage limit listed on the data sheet.

5.2.3 Clamping Performance

The voltage clamping performance of each energy dissipation circuit was simulated as detailed in Chapter 4.5.3 Simulation Setup and Test. The two main aspects of the clamping performance under consideration are the maximum voltage experienced during the clamping transient and the transient time period. The maximum transient voltage will also be compared to the voltage clamping limits listed in the manufacturer data sheet. Results consistent with manufacturer provided data as well as a fast transient time is desirable.

As documented in the previous chapter, TVS 7 and TVS 8 failed to appropriately clamp voltage during a transient. This failure of TVS 7 and TVS 8 models coupled with the previously mentioned breaking voltage discrepancies may indicate inaccuracies with the simulation models. With this in mind, any further investigation using these models will be abandoned and they will be removed from the selection pool. The remaining six TVS diode models, TVS 1 through TVS 6, being considered for the energy dissipation circuit are from Littelfuse Inc. An overview of their voltage clamping behavior can be seen in Figure 5.17.

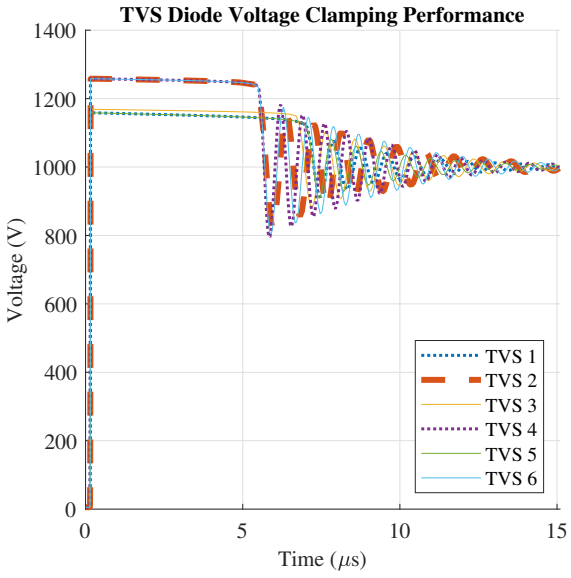


Figure 5.17: Energy Dissipation Circuit Voltage Clamping Performance in a Bidirectional SSCB Circuit

In Figure 5.18, the time scale has been reduced to highlight the clamping knee. The plot line color and style of each model has been maintained from the previous figure with an overview of the clamping performance, Figure 5.17, as well as the figures showcasing the current versus voltage comparison, Figure 5.16.

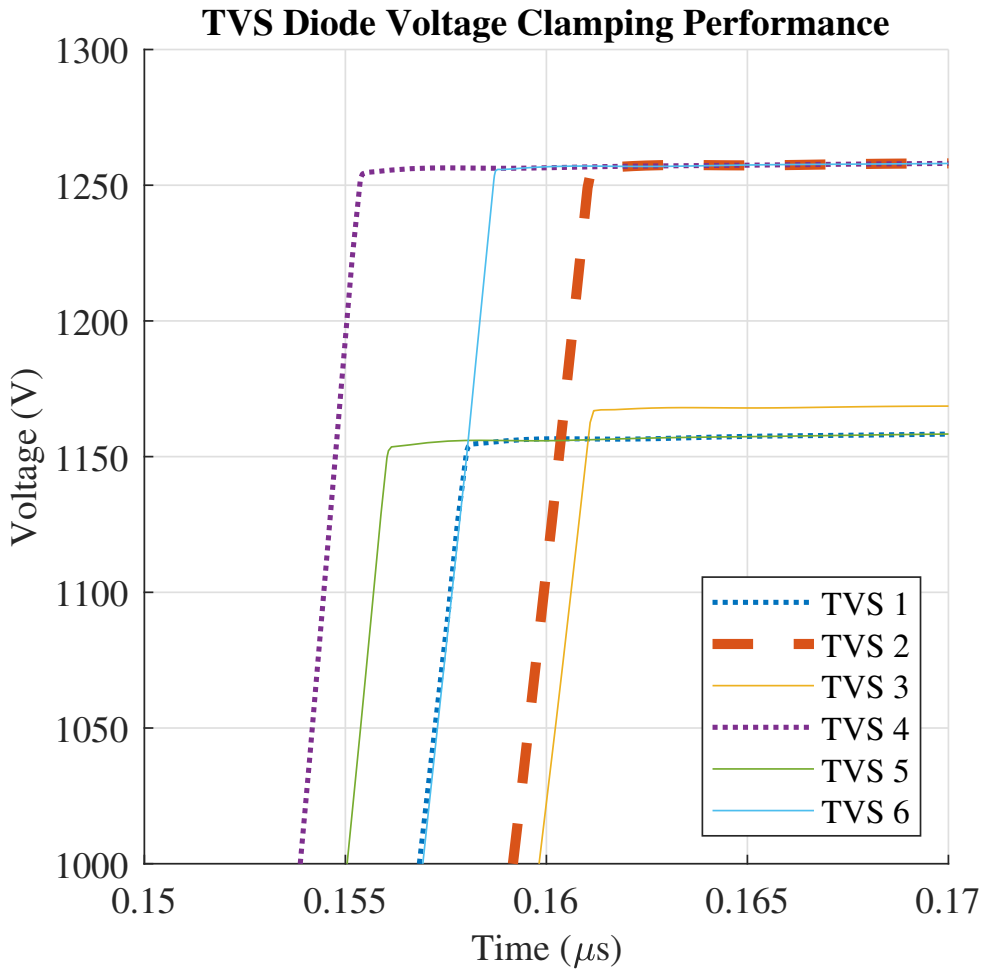


Figure 5.18: Energy Dissipation Circuit Voltage Clamping Performance in a Bidirectional SSCB Circuit

As can be seen in Figure 5.18, there are three groups of voltage clamp responses. This coincides with the three groups of individual stand-off voltage ratings: $530V$ (TVS 3), $380V$ (TVS 2, TVS 4, TVS 6), and $170V$ (TVS 1, TVS 5). Each one of these groups were able to clamp voltage above the operating voltage of $1000V$ and below the more limited clamp limit of $1440V$.

There is only one model rated at a stand-off voltage of $530V$, TVS 3. Its maximum peak voltage was $1167V$. There is nothing overtly negative about the clamping performance of this model. Though, as a reminder the solid-state switch to be used is rated for $1200V$.

There are three models rated at a stand-off voltage of $380V$, TVS 2, TVS 6, and TVS 4. The maximum peak voltage of this group was $1256V$. These models did clamp above the rated switch voltage, which would allow the SSCB to be operated at the rated switch voltage if necessary.

There are two models rated at a stand-off voltage of $170V$, TVS 1 and TVS 5. The maximum peak voltage of this group was $1154V$. These models performed well and like the previous $530V$ model have a clamping voltage below the switch rated voltage.

Each one of these models performed well. Though, two groups with stand-off voltages of $530V$ and $170V$ had clamping voltages below the switch rated voltage. There is nothing in the selection criteria restricting these models, though using them would mean the energy dissipation circuit is the limiting factor in the SSCB. A discreet view of the clamping limits can be seen in Table 5.10.

Designation	$\mathbf{n_{EDC}}$	$\mathbf{V_{CL}}$ (V)	$\mathbf{V_{CL,EDC}}$ (V)	$\mathbf{V_{CL,SIM}}$ (V)
TVS 1	6	170	1560	1154
TVS 2	3	380	1560	1256
TVS 3	2	750	1500	1167
TVS 4	3	380	1560	1254
TVS 5	6	170	1560	1154
TVS 6	3	380	1560	1255

Table 5.10: Energy Dissipation Circuit Clamping Voltage Comparison

The results displayed in Table 5.10 are accompanied with the estimated clamping limit of the energy dissipation circuit, the individual device clamping limit, and the number of devices required in series. Taking into account the switch rated voltage and the number of devices required for each configuration, the three models rated at a stand-off voltage of $380V$ (TVS 2, TVS 4, TVS 6) are prime candidates.

Another aspect of performance to consider is the transient time, or how fast does the energy dissipation circuit reach the clamping limit. A faster reaction time would better protect any switch installed in parallel. The transient times were gathered from the clamping performance figure. The starting point for measurement was the same for each model, being the origin, with the end point being the clamping knee point.

Designation	$\mathbf{n_{EDC}}$	$\mathbf{t_{CL,SIM}}$ (μs)
TVS 1	6	0.1581
TVS 2	3	0.1611
TVS 3	2	0.1612
TVS 4	3	0.1554
TVS 5	6	0.1561
TVS 6	3	0.1587

Table 5.11: Energy Dissipation Circuit Clamp Timing Comparison

The results displayed in Table 5.11 are accompanied with the number of devices required in series. The difference between the fastest clamping time and the slowest clamping time is less than $6ns$. This means that all the TVS diode models performed well and had similar transient response times.

Reviewing the overall clamping performance of each of the TVS diode models did give insight into which models would performance best. Taking into account the previous results that acknowledged the switch rated voltage of $1200V$. The three models rated at a stand-off voltage of $380V$, TVS 4, TVS 6, and TVS 2, contained the first, fourth, and fifth fastest response.

The fastest response in the selection pool being TVS 4 with a response time of $0.1554\mu s$. A second choice would be TVS 6 with a response time of $0.1587\mu s$. The response time difference between these two models is $3.3ns$, making either of these models a good candidate.

As a reminder the model number difference between these two devices is related to the pulse peak current rating. A model number with AK3 has a pulse peak current rating of $3000A$ and a model number with AK6 has a pulse peak current rating of $6000A$.

5.2.4 Energy Absorption

The energy absorption performance of each energy dissipation circuit was simulated as detailed in Chapter 4.5.3 Simulation Setup and Test. The aspect of the energy absorption performance under consideration is the maximum energy absorbed during the clamping transient. The more energy absorbed by the energy dissipation circuit, the less that will need to be absorbed by other components including the solid-state switch. An overview of the energy absorption of each device during a clamping performance can be seen in Figure 5.19.

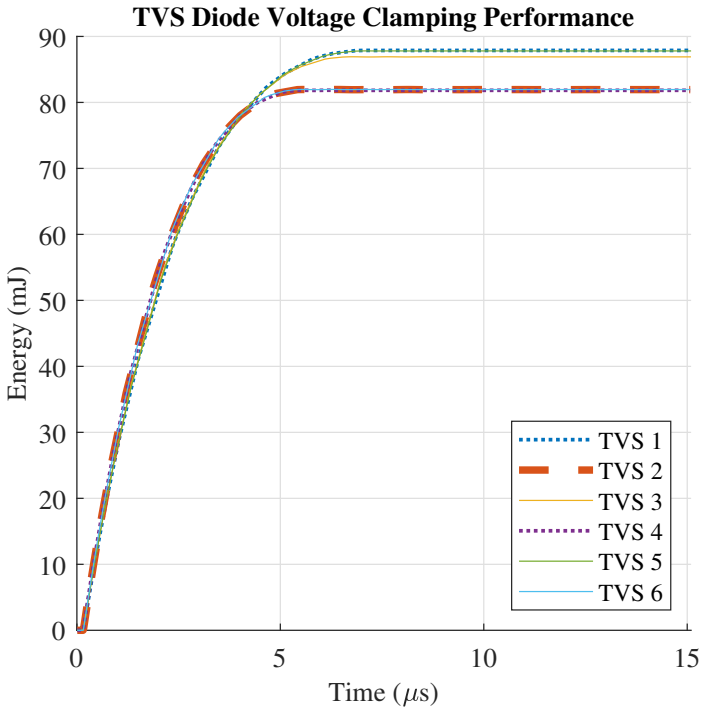


Figure 5.19: Energy Dissipation Circuit Energy Absorption Performance in a Bidirectional SSCB Circuit

Measuring the amount of energy absorbed by the energy dissipation circuit at the moment a clamping transient occurred was accomplished using the data collection method as detailed in Chapter 4.5.3 Simulation Setup and Test. The data was normalized to the just before the switching transient, thus capturing the summation of energy as it was being absorbed by the energy dissipation circuit.

In Figure 5.20 the time scale has been reduced highlighting the final transient energy values. The plot line color and style of each model has been maintained from the previous figure with an overview of the energy absorption, Figure 5.19, as well as the figures showcasing the voltage clamping performance and current versus voltage comparison.

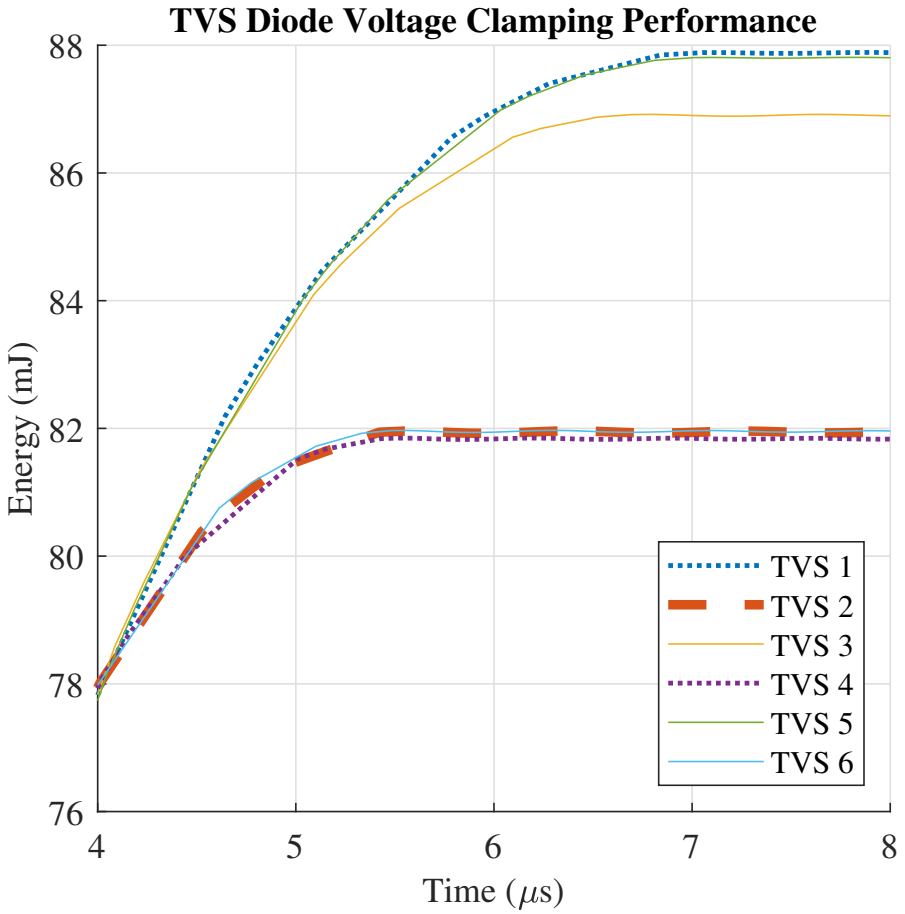


Figure 5.20: Energy Dissipation Circuit Energy Absorption Performance in a Bidirectional SSCB Circuit

As can be seen in Figure 5.20, there are three groups of energy absorption final values. This coincides with the three groups of individual stand-off voltage ratings: $530V$, $380V$, and $170V$. Each one of these groups were able to absorb over $80mJ$ of energy within $8\mu s$ of the clamping transient.

Each one of these models performed well. Though, two groups with stand-off voltages of $530V$ and $170V$ had higher values of energy absorbed and the group with stand-off voltages of $380V$ had the fastest transient time. A discreet view of the final transient energy values can be seen in Table 5.12.

Designation	$\mathbf{n_{EDC}}$	$\mathbf{w_{EDC}}$ (mJ)
TVS 1	6	87.83
TVS 2	3	81.94
TVS 3	2	86.91
TVS 4	3	81.82
TVS 5	6	87.80
TVS 6	3	82.12

Table 5.12: Energy Dissipation Circuit Clamp Energy Comparison

There is only one model rated at a stand-off voltage of $530V$; TVS 3. The final transient energy absorbed by this model was $86.91mJ$. The energy transient took approximately $6.5\mu s$.

There are three models rated at a stand-off voltage of $380V$; TVS 2, TVS 6, and TVS 4. The final transient energy absorbed by this group was approximately $82mJ$. The energy transient took approximately $5.5\mu s$, which was the fastest time in the selection pool.

There are two models rated at a stand-off voltage of $170V$; TVS 1 and TVS 5. The final transient energy absorbed by this group was approximately $88mJ$. The energy transient took approximately $7\mu s$, which was the slowest time in the selection pool.

The models rated at a stand-off voltage of $380V$ absorbed $6mJ$ of energy less than the models rated at a stand-off voltage of $170V$, but the transient experienced was $1.5\mu s$ faster. Again, each one of these models performed well and a clear delineation can not be made using energy absorption performance alone.

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

The following sections will fully explain the conclusions reached from the earlier gathered results as well as other possible outcomes. These conclusion will contain final decisions on primary components to be used as well as explain the reasoning in the decision-making process.

6.1 Conclusions Format

The conclusions reached in this chapter will be divided into three main sections: Energy Dissipation Circuit, Solid-State Switch, and Future Work. The EDC section will cover the conclusions from the comparison of cost, activation range, and clamping performance. The Solid-State Switch section will include detailed discussion on unbiased comparison and hypothetical scenarios.

6.2 Energy Dissipation Circuit

In review, the results from the comparisons have been complied into Table 6.1. These results will be discussed including any insights not previously covered, ending with a single device being chosen for the EDC. Following this discussion values included in Table 6.1 will be normalized and reviewed to confirm component choice.

Criteria	**TVS 1**	**TVS 2**	**TVS 3**	**TVS 4**	**TVS 5**	**TVS 6**	**TVS 7**	**TVS 8**
n_{edc}	6	3	2	3	6	3	3	3
$V_{active,EDC}(V)$	540	420	440	420	540	420	420	420
$V_{cl,EDC}(V)$	1560	1560	1500	1560	1560	1560	1560	1560
$V_{cl,SIM}(V)$	1154	1256	1167	1254	1154	1255	1571	1575
$Price_{unit}(\$)$	57.75	100.75	142.00	50.12	47.06	77.60	93.29	46.41
$Price_{total}(\$)$	346.50	302.25	284.00	150.36	282.36	232.50	279.87	139.23
$Price_{watt}(\$/kW)$	0.02221	0.01938	0.01893	0.03213	0.03017	0.02484	0.01794	0.02975
$t_{cl}(\mu s)$	0.1581	0.1611	0.1612	0.1554	0.1561	0.1587	-	-
$w_{EDC}(mJ)$	87.83	81.94	86.91	81.82	87.80	82.12	-	-

Table 6.1: Energy Dissipation Circuit Comparison Criteria

The cost comparison of the energy dissipation circuits resulted with TVS 6 being the best overall. Though this model did not excel in any one metric, but was the best middle ground between unit price, total price, and price per power rating.

The activation range comparison of the energy dissipation circuits resulted with TVS 7 and TVS 8 being the best overall. These models were later eliminated from the selection pool. The next models to be considered were TVS 2, TVS 6, and TVS 4. These models provided the same activation range as the Bourns Inc. models and have a breaking voltage range only $3V$ larger.

The clamping performance comparison of the energy dissipation circuits resulted with TVS 2, TVS 6, and TVS 4 being best overall, meeting a stretch requirement of the switch rated voltage. In particular the TVS 4 performed the best in the group in terms of clamping speed, followed by the TVS 6.

The energy absorption comparison of the energy dissipation circuits resulted with TVS 1 and TVS 5 performing best overall. These models absorbed approximately $6mJ$ more than the other models, but did have a longer transient.

Considering each of the results concludes with the TVS diode model TVS 6 having the best performance over all the comparison criteria considered.

6.2.1 EDC Normalized Comparison

The normalization of the EDC result values can be accomplished using Equation 6.1 and Equation 6.2. Completing this process will result in criteria values being the most desirable as a value of one. If a criteria is more desirable to be higher than Equation 6.1 will be applied. If a criteria is more desirable to be lower than Equation 6.2 will be applied.

$$Value_{normalized} = 1 - ((MAX - Value)/1000) \tag{6.1}$$

$$Value_{normalized} = 1 - ((Value - MIN)/1000) \tag{6.2}$$

The results from the normalization process can be found in Table 6.2, with the models from Bourns Inc. (TVS 7 and TVS 8) removed. A definitive answer as to which EDC performed the best out of these criteria can be determined by summing the values in each column. The highest total value means that particular model performed best overall considered criteria. Noting that each criteria is equally weighted.

Criteria	***TVS 1***	***TVS 2***	***TVS 3***	***TVS 4***	***TVS 5***	***TVS 6***
n_{edc}	0.996	0.999	1	0.999	0.996	0.999
$V_{active,EDC}(V)$	0.88	1	0.92	1	0.88	1
$V_{cl,EDC}(V)$	1	1	1	1	1	1
$V_{cl,SIM}(V)$	1	0.898	0.987	0.9	1	0.899
$Price_{unit}(\$)$	0.98866	0.94566	0.90441	0.99629	0.99935	0.96881
$Price_{total}(\$)$	0.79273	0.83698	0.85523	0.98887	0.85687	0.90673
$Price_{watt}(\$/kW)$	0.999996	0.999999	0.999999	0.999986	0.999988	0.999993
$t_{cl}(\mu s)$	0.999997	0.999994	0.999994	1	0.999999	0.999997
$w_{EDC}(mJ)$	1	0.99411	0.99908	0.99399	0.99997	0.99429
Total	8.657383	8.673743	8.665713	8.878136	8.732177	8.76782

Table 6.2: Energy Dissipation Circuit Comparison Criteria Normalized

Using the unbiased normalization review process resulted in the TVS 4 model being the best overall, not the previously determined TVS 6 model. Though, the TVS 6 model was the next best with a difference slightly larger than 0.1 in totals. This result reiterates the importance of an unbiased review process. Also, this result highlights an opportunity for further review using weighted values for criteria deemed more desirable.

6.3 Solid-State Switch

At the completion of the analysis, design, and simulation phases, there was twenty eight different criteria to compare for each solid-state switch. These criteria included efficiencies at different temperatures, loads, and configurations. As well as cost comparisons and thermal considerations. Ensuring an unbiased comparison over this wide range of criteria would be difficult [100]. In an attempt to alleviate this burden, an decision matrix will be developed.

6.3.1 Decision Matrix

The decision matrix will consist of rows of criteria and columns of devices. The value at the intersection of the criteria row and device column will correspond to the results data gathered for that device in that criteria. These values can be seen in Table 6.3. The result values gathered will need to be further normalized and then weighted as deemed appropriate by the application.

6.3.2 Normalized Values

Similar to the process completed for the EDC, the normalization of the result values can be accomplished using Equation 6.3 and Equation 6.4. The difference being the number of digits being applied. Remaining the same is that all of the result values in a criteria are normalized to the most desirable value. For example, when considering efficiencies the most desirable value is 100%, therefore applying Equation 6.3 will result in the highest efficiency in that criteria being one. The difference from the highest value to the lowest will remain. If the criteria is most desirable to have a small value, for instance price, then Equation 6.4 will be used resulting in the lowest value being one. Again, the difference in values is maintained.

$$Value_{normalized} = 1 - ((MAX - Value)/100) \tag{6.3}$$

$$Value_{normalized} = 1 - ((Value - MIN)/100) \tag{6.4}$$

The normalized values of the results values from Table 6.3 can be seen in Table 6.4. A decision can be made using these normalized values, by summing all the normalized values in the device column. Due to the normalization process the most desirable value of any criteria is one, with all other values being less than one. Therefore, the largest total value is the best performing device overall. Keeping in mind that each of the normalized values have equal bearing on the final total value.

Table 6.3: Simulation Generated Values with Criteria

Criteria			Device 1	Device 2	Device 3	Device 4	Device 5	Device 6	Device 7	Device 8	Device 9	Device 10	Device 11
Unidirectional Efficiency (%)	-55°C	100 Ω	99.9	99.92	99.93	99.89	99.93	99.92	99.93	99.93	99.92	99.86	99.87
		34 Ω	99.66	99.69	99.8	99.65	99.78	99.76	99.78	99.78	99.76	99.8	99.79
		30 Ω	99.6	99.63	99.77	99.6	99.76	99.73	99.75	99.75	99.72	99.78	99.78
	25°C	100 Ω	99.94	99.92	99.94	99.92	99.92	99.92	99.92	99.93	99.93	99.83	99.85
		34 Ω	99.8	99.74	99.82	99.74	99.77	99.75	99.77	99.76	99.78	99.74	99.76
		30 Ω	99.76	99.7	99.79	99.7	99.74	99.72	99.74	99.72	99.75	99.73	99.74
	150°C	100 Ω	99.9	99.89	99.92	99.9	99.85	99.84	99.86	99.87	99.91	99.81	99.82
		34 Ω	99.68	99.68	99.74	99.64	99.55	99.53	99.57	99.57	99.71	99.67	99.64
		30 Ω	99.63	99.63	99.7	99.58	99.49	99.46	99.51	99.51	99.66	99.64	99.6
	175°C	100 Ω	99.89	99.89	99.91	99.87	99.83	99.83	99.84	99.84	99.9	99.8	99.82
		34 Ω	99.64	99.65	99.7	99.53	99.51	99.48	99.53	99.51	99.65	99.65	99.65
		30 Ω	99.58	99.61	99.66	99.44	99.44	99.41	99.46	99.44	99.59	99.62	99.61
Bidirectional Efficiency (%)	-55°C	100 Ω	99.83	99.84	99.87	99.78	99.85	99.84	99.85	99.87	99.85	99.67	99.74
		34 Ω	99.44	99.47	99.6	99.39	99.57	99.53	99.57	99.57	99.54	99.48	99.6
		30 Ω	99.35	99.38	99.54	99.31	99.51	99.47	99.51	99.5	99.48	99.44	99.58
	25°C	100 Ω	99.89	99.85	99.88	99.84	99.85	99.83	99.85	99.86	99.87	99.61	99.71
		34 Ω	99.66	99.52	99.64	99.51	99.55	99.51	99.55	99.53	99.59	99.39	99.55
		30 Ω	99.61	99.45	99.58	99.44	99.49	99.44	99.49	99.46	99.53	99.35	99.52
	150°C	100 Ω	99.81	99.79	99.84	99.79	99.7	99.68	99.71	99.73	99.83	99.58	99.67
		34 Ω	99.44	99.37	99.5	99.35	99.11	99.07	99.16	99.26	99.45	99.29	99.41
		30 Ω	99.36	99.28	99.44	99.27	99	98.95	99.05	99.19	99.36	99.24	99.36
	175°C	100 Ω	99.78	99.77	99.82	99.74	99.67	99.65	99.69	99.69	99.8	99.58	99.68
		34 Ω	99.36	99.32	99.46	99.2	99.03	98.98	99.08	99.19	99.34	99.27	99.43
		30 Ω	99.28	99.23	99.39	99.09	98.9	98.85	98.96	99.11	99.24	99.21	99.37
Price_{Unit} ($)			15.45	15.12	25.13	18.23	12.6	12.6	14.39	16.67	24.05	5.7	5.11
Price_{Amp} ($/A)			0.4292	0.3877	0.6792	0.588	0.3761	0.3818	0.4171	0.4631	0.8017	0.19	0.1703
$R_{\theta,sa}$ (°C/W)			1.2446	0.9821	0.9304	0.7721	1.3195	1.0921	1.3332	1.0321	0.6946	0.7268	1.0692
$R_{\theta,sa,Limited}$ (°C/W)			1.4029	1.1009	1.0492	0.8909	1.4543	1.2109	1.4689	1.1509	0.8213	0.8304	1.1891

Table 6.4: Normalized Values with Criteria

Criteria			Device 1	Device 2	Device 3	Device 4	Device 5	Device 6	Device 7	Device 8	Device 9	Device 10	Device 11
Unidirectional Efficiency	-55°C	100 Ω	0.9997	0.9999	1	0.9996	1	0.9999	1	1	0.9999	0.9993	0.9994
		34 Ω	0.9986	0.9989	1	0.9985	0.9998	0.9996	0.9998	0.9998	0.9996	1	0.9999
		30 Ω	0.9982	0.9985	0.9999	0.9982	0.9998	0.9995	0.9997	0.9997	0.9994	1	1
	25°C	100 Ω	1	0.9998	1	0.9998	0.9998	0.9998	0.9998	0.9999	0.9999	0.9989	0.9991
		34 Ω	0.9998	0.9992	1	0.9992	0.9995	0.9993	0.9995	0.9994	0.9996	0.9992	0.9994
		30 Ω	0.9997	0.9991	1	0.9991	0.9995	0.9993	0.9995	0.9993	0.9996	0.9994	0.9995
	150°C	100 Ω	0.9998	0.9997	1	0.9998	0.9993	0.9992	0.9994	0.9995	0.9999	0.9989	0.999
		34 Ω	0.9994	0.9994	1	0.999	0.9981	0.9979	0.9983	0.9983	0.9997	0.9993	0.999
		30 Ω	0.9993	0.9993	1	0.9988	0.9979	0.9976	0.9981	0.9981	0.9996	0.9994	0.999
	175°C	100 Ω	0.9998	0.9998	1	0.9996	0.9992	0.9992	0.9993	0.9993	0.9999	0.9989	0.9991
		34 Ω	0.9994	0.9995	1	0.9983	0.9981	0.9978	0.9983	0.9981	0.9995	0.9995	0.9995
		30 Ω	0.9992	0.9995	1	0.9978	0.9978	0.9975	0.998	0.9978	0.9993	0.9996	0.9995
Bidirectional Efficiency	-55°C	100 Ω	0.9996	0.9997	1	0.9991	0.9998	0.9997	0.9998	1	0.9998	0.998	0.9987
		34 Ω	0.9984	0.9987	1	0.9979	0.9997	0.9993	0.9997	0.9997	0.9994	0.9988	1
		30 Ω	0.9977	0.998	0.9996	0.9973	0.9993	0.9989	0.9993	0.9992	0.999	0.9986	1
	25°C	100 Ω	1	0.9996	0.9999	0.9995	0.9996	0.9994	0.9996	0.9997	0.9998	0.9972	0.9982
		34 Ω	1	0.9986	0.9998	0.9985	0.9989	0.9985	0.9989	0.9987	0.9993	0.9973	0.9989
		30 Ω	1	0.9984	0.9997	0.9983	0.9988	0.9983	0.9988	0.9985	0.9992	0.9974	0.9991
	150°C	100 Ω	0.9997	0.9995	1	0.9995	0.9986	0.9984	0.9987	0.9989	0.9999	0.9974	0.9983
		34 Ω	0.9994	0.9987	1	0.9985	0.9961	0.9957	0.9966	0.9976	0.9995	0.9979	0.9991
		30 Ω	0.9992	0.9984	1	0.9983	0.9956	0.9951	0.9961	0.9975	0.9992	0.998	0.9992
	175°C	100 Ω	0.9996	0.9995	1	0.9992	0.9985	0.9983	0.9987	0.9987	0.9998	0.9976	0.9986
		34 Ω	0.999	0.9986	1	0.9974	0.9957	0.9952	0.9962	0.9973	0.9988	0.9981	0.9997
		30 Ω	0.9989	0.9984	1	0.997	0.9951	0.9946	0.9957	0.9972	0.9985	0.9982	0.9998
Price_{Unit}			0.8966	0.8999	0.7998	0.8688	0.9251	0.9251	0.9072	0.8844	0.8106	0.9941	1
Price_{Amp}			0.997411	0.997826	0.994911	0.995823	0.997942	0.997885	0.997532	0.997072	0.993686	0.999803	1
$\text{R}_{\theta,sa}$			0.999114	0.996489	0.995972	0.994389	0.999863	0.997589	1	0.996989	0.993614	0.993936	0.99736
$\text{R}_{\theta,sa,Limited}$			0.99934	0.99632	0.995803	0.99422	0.999854	0.99742	1	0.99682	0.993524	0.993615	0.997202
Total			27.87687	27.86924	27.78539	27.82143	27.88726	27.87599	27.87253	27.84748	27.77952	27.94835	27.97656

6.3.3 Weighted Values

If specific criteria were more or less desirable in a particular application the normalized values could be weighted as such. The weighting of the normalized result values can be accomplished using Equation 6.5, Equation 6.6, and Equation 6.7. The weighted portion is the single portion of the total criteria.

$$Weight_{portion} = 1/Total_{criteria} \tag{6.5}$$

In this case, with twenty eight different criteria a single criteria would normally have $1/28^{th}$ of the total. This value can be biased or skewed to favor specific criteria over others, by using a weight factor.

$$Weight_{factor} = Weight_{portion} \times percentage \tag{6.6}$$

The weight factor is simply a percentage of the weight portion. For example, if a criteria is more critical for the desired application it can be weighted at 200%, but other criteria must be reduced to maintain the same total percentage. The total percentage of all the criteria must be equal to the number of different criteria; in this case 28. Applying this weight factor to the normalized value will result in a weighted value.

$$Value_{weighted} = Value_{normalized} \times Weight_{factor} \tag{6.7}$$

An example of using weighted values from the normalized values in Table 6.4 can be seen in Table 6.5. Similarly with the normalized values, a decision can be made using the weighted values summing all the weighted values in the device column. As of a result of the normalization and weighting process the highest weighted value achievable will be one. Therefore, the total closest to one will result in the best performing device overall. In this example the best overall changed from Device 11 in normalized values to Device 3. This is because the cost criteria was deemed less important.

Criteria		Percentage	Device 1	Device 2	Device 3	Device 4	Device 5	Device 6	Device 7	Device 8	Device 9	Device 10	Device 11
Unidirectional Efficiency	-55°C	100 Ω	1.0	0.035704	0.035711	0.035714	0.0357	0.035714	0.035711	0.035714	0.035714	0.035711	0.035689	0.035693
		34 Ω	1.0	0.035664	0.035675	0.035714	0.035661	0.035707	0.0357	0.035707	0.035707	0.0357	0.035714	0.035711
		30 Ω	0.9	0.032085	0.032095	0.03214	0.032085	0.032136	0.032127	0.032133	0.032133	0.032124	0.032143	0.032143
		100 Ω	1.6	0.057143	0.057131	0.057143	0.057131	0.057131	0.057131	0.057131	0.057137	0.057137	0.05708	0.057091
	25°C	34 Ω	1.6	0.057131	0.057097	0.057143	0.057097	0.057114	0.057103	0.057114	0.057109	0.05712	0.057097	0.057109
		30 Ω	0.9	0.032133	0.032114	0.032143	0.032114	0.032127	0.03212	0.032127	0.03212	0.03213	0.032124	0.032127
		100 Ω	1.6	0.057131	0.057126	0.057143	0.057131	0.057103	0.057097	0.057109	0.057114	0.057137	0.05708	0.057086
	150°C	34 Ω	1.6	0.057109	0.057109	0.057143	0.057086	0.057034	0.057023	0.057046	0.057046	0.057126	0.057103	0.057086
		30 Ω	0.9	0.03212	0.03212	0.032143	0.032104	0.032075	0.032066	0.032082	0.032082	0.03213	0.032124	0.032111
		100 Ω	0.9	0.032136	0.032136	0.032143	0.03213	0.032117	0.032117	0.03212	0.03212	0.03214	0.032108	0.032114
	175°C	34 Ω	0.9	0.032124	0.032127	0.032143	0.032088	0.032082	0.032072	0.032088	0.032082	0.032127	0.032127	0.032127
		30 Ω	0.8	0.028549	0.028557	0.028571	0.028509	0.028509	0.0285	0.028514	0.028509	0.028551	0.02856	0.028557
Unidirectional Efficiency	-55°C	100 Ω	1.0	0.0357	0.035704	0.035714	0.035682	0.035707	0.035704	0.035707	0.035714	0.035707	0.035643	0.035668
		34 Ω	1.0	0.035657	0.035668	0.035714	0.035639	0.035704	0.035689	0.035704	0.035704	0.035693	0.035671	0.035714
		30 Ω	0.9	0.032069	0.032079	0.03213	0.032056	0.03212	0.032108	0.03212	0.032117	0.032111	0.032098	0.032143
		100 Ω	1.6	0.057143	0.05712	0.057137	0.057114	0.05712	0.057109	0.05712	0.057126	0.057131	0.056983	0.05704
	25°C	34 Ω	1.6	0.057143	0.057063	0.057131	0.057057	0.05708	0.057057	0.05708	0.057069	0.057103	0.056989	0.05708
		30 Ω	0.9	0.032143	0.032091	0.032133	0.032088	0.032104	0.032088	0.032104	0.032095	0.032117	0.032059	0.032114
		100 Ω	1.6	0.057126	0.057114	0.057143	0.057114	0.057063	0.057051	0.057069	0.05708	0.057137	0.056994	0.057046
	150°C	34 Ω	1.6	0.057109	0.057069	0.057143	0.057057	0.05692	0.056897	0.056949	0.057006	0.057114	0.057023	0.057091
		30 Ω	0.9	0.032117	0.032091	0.032143	0.032088	0.032001	0.031985	0.032018	0.032063	0.032117	0.032079	0.032117
		100 Ω	0.9	0.03213	0.032127	0.032143	0.032117	0.032095	0.032088	0.032101	0.032101	0.032136	0.032066	0.032098
	175°C	34 Ω	0.9	0.032111	0.032098	0.032143	0.032059	0.032005	0.031989	0.032021	0.032056	0.032104	0.032082	0.032133
		30 Ω	0.8	0.02854	0.028526	0.028571	0.028486	0.028431	0.028417	0.028449	0.028491	0.028529	0.02852	0.028566
Price_{Unit}		0.1		0.003202	0.003214	0.002856	0.003103	0.003304	0.003304	0.00324	0.003159	0.002895	0.00355	0.003571
Price_{Amp}		0.1		0.003562	0.003564	0.003553	0.003557	0.003564	0.003564	0.003563	0.003561	0.003549	0.003571	0.003571
$\text{R}_{\theta,sa}$		0.1		0.003568	0.003559	0.003557	0.003551	0.003571	0.003563	0.003571	0.003561	0.003549	0.00355	0.003562
$\text{R}_{\theta,sa,Limited}$		0.3		0.010707	0.010675	0.010669	0.010652	0.010713	0.010687	0.010714	0.01068	0.010645	0.010646	0.010684
Total		28.0		0.999056	0.998758	0.999165	0.998258	0.998352	0.998067	0.998415	0.998454	0.99877	0.998471	0.999152

Table 6.5: Weighted Values with Criteria

6.4 Application Notes

The following subsections detail three scenarios in which different operating criteria have been deemed more desirable. These criteria have been weighted accordingly, resulting in a different solid-state switch model that would be best suited. In an attempt at simplicity, the weighted values table has been divided into two separate tables. One table will display the weighted percentages applied to the criteria. Another table will showcase the final totals of each model, with a gradient heat map applied. The heat map will indicate the most desirable model in green, the least desirable in red, and yellow will indicate the mid-grade.

6.4.1 Scenario 1

In this scenario, a SSCB is a component in a power distribution network. Steady state operations in this system dictate specific operating parameters and requires efficiency to be heavily weighted at almost any cost or thermal load. Specifically, the bidirectional efficiency of the SSCB operating at $150°C$ with a load demand at or exceeding the upper limit. These desired parameters can be seen in Table 6.6.

Criteria			**Unidirectional**	**Bidirectional**
Efficiency	$-55°C$	100Ω	1	1
		34Ω	1	1
		30Ω	1	1
	$25°C$	100Ω	1	1
		34Ω	1	1
		30Ω	1	1
	$150°C$	100Ω	1	1
		34Ω	1	2.8
		30Ω	1	2.8
	$175°C$	100Ω	1	1
		34Ω	1	1
		30Ω	1	1
Price_{Unit}			0.1	
Price_{Amp}			0.1	
$\text{R}_{\theta,sa}$			0.1	
$\text{R}_{\theta,sa,Limited}$			0.1	
Total			12.2	15.8
			28	

Table 6.6: Weighted Criteria for Scenario 1

The results from applying the weighted criteria for Scenario 1 can be seen in Table 6.7, as well as the equally weighted normalized total and change between the two totals. In this scenario there was a total difference between the models of 0.00137. The model with the highest total was Device 3. Followed by Device 11 at a difference of less than 0.003%.

Designation	Normalized Total	Weighted Total	Difference
Device 1	0.995602	0.999001	+0.003399
Device 2	0.995330	0.998733	+0.003403
Device 3	0.992335	0.999198	+0.006863
Device 4	0.993623	0.998212	+0.004589
Device 5	0.995974	0.998116	+0.002142
Device 6	0.995571	0.997828	+0.002257
Device 7	0.995448	0.998208	+0.002760
Device 8	0.994553	0.998337	+0.003784
Device 9	0.992126	0.998792	+0.006666
Device 10	0.998156	0.998449	+0.000294
Device 11	0.999163	0.999171	+0.000008

Table 6.7: Weighted Totals for Scenario 1

Referring back to the normalized values, Table 6.4, Device 3 had the second highest unit price and the best overall efficiency. This initial high cost would be an acceptable burden over the loss in efficiency. Even if any cost savings did occur from using a cheaper device they would be overrun by the constant and continuing cost of a lowered transmission efficiency.

6.4.2 Scenario 2

Establishing a single operating criteria allows for a better engineered solution, but this is not always feasible. In this scenario, a SSCB is a component in a local distribution network of a microgrid. The SSCB is a part of the initial install of the system that is located in the northern hemisphere temperature zone. In this case, there are two driving factors: potential for colder operations and startup costs. The weighted criteria can be skewed in more than one direction to favor devices that would operate well in a colder environment while respecting the cost of each device. These desired parameters can be

seen in Table 6.8, in which operating temperatures of 150°C and 175°C have been reduced. As well as cost considerations have been increased.

Criteria			**Unidirectional**	**Bidirectional**
Efficiency	-55°C	100Ω	1	1
		34Ω	1	1
		30Ω	1	1
	25°C	100Ω	1	1
		34Ω	1	1
		30Ω	1	1
	150°C	100Ω	0.9	0.9
		34Ω	0.9	0.9
		30Ω	0.8	0.8
	175°C	100Ω	0.8	0.8
		34Ω	0.8	0.8
		30Ω	0.8	0.8
Price_{Unit}			2	
Price_{Amp}			2	
$\text{R}_{\theta,sa}$			1	
$\text{R}_{\theta,sa,Limited}$			1	
Total			14	14
			28	

Table 6.8: Weighted Criteria for Scenario 2

The results from applying the weighted criteria for Scenario 2 can be seen in Table 6.9, as well as the equally weighted normalized total and change between the two totals. In this scenario the model with the highest total was Device 11. Followed by Device 10 at a difference of less than 0.118%.

Designation	Normalized Total	Weighted Total	Difference
Device 1	0.995602	0.991863	-0.003739
Device 2	0.995330	0.991737	-0.003593
Device 3	0.992335	0.985003	-0.007332
Device 4	0.993623	0.988896	-0.004726
Device 5	0.995974	0.993411	-0.002562
Device 6	0.995571	0.993029	-0.002543
Device 7	0.995448	0.99221	-0.003237
Device 8	0.994553	0.990454	-0.004099
Device 9	0.992126	0.985178	-0.006948
Device 10	0.998156	0.998037	-0.000118
Device 11	0.999163	0.999219	+0.000056

Table 6.9: Weighted Totals for Scenario 2

Referring back to the normalized values, Table 6.4, Device 11 had the best unit price, best price per amp, and operated well at lower temperatures. The large difference in the total weighted values came down to the cost considerations. Most of the devices operate well overall, but the best efficiency has a premium cost.

6.4.3 Scenario 3

In this final scenario, a SSCB is installed as the isolation between a microgrid and a home. The SSCB will be inside the home and will not be operated regularly. In fact the owner may not understand how to operate the device and would prefer a "set it and forget" solution. In this case, the weighted criteria for the extreme temperature ranges would be suppressed. Also, thermal operating properties would be more of a concern. In that a simplified cooling apparatus that would require little to no maintenance would be preferred. These desired parameters can be seen in Table 6.10, in which operating temperatures of -55°C and 175°C have been reduced. As well as the reduction of cost considerations and an increase in thermal related criteria.

Criteria			**Unidirectional**	**Bidirectional**
Efficiency	-55°C	100Ω	0.5	0.5
		34Ω	0.5	0.5
		30Ω	0.5	0.5
	25°C	100Ω	1.5	1.5
		34Ω	1.5	1.5
		30Ω	1.5	1.5
	150°C	100Ω	1.5	1.5
		34Ω	1.5	1.5
		30Ω	1.5	1.5
	175°C	100Ω	0.5	0.5
		34Ω	0.5	0.5
		30Ω	0.5	0.5
Price_{Unit}			0.1	
Price_{Amp}			0.1	
$\text{R}_{\theta,sa}$			1.9	
$\text{R}_{\theta,sa,Limited}$			1.9	
Total			14	14
			28	

Table 6.10: Weighted Criteria for Scenario 3

The results from applying the weighted criteria for Scenario 3 can be seen in Table 6.11, as well as the equally weighted normalized total and change between the two totals.

In this scenario the model with the highest total was Device 1. Followed by Device 11 at a difference of less than 0.023%.

Designation	Normalized Total	Weighted Total	Difference
Device 1	0.995602	0.999106	+0.003504
Device 2	0.995330	0.998399	+0.003069
Device 3	0.992335	0.998668	+0.006332
Device 4	0.993623	0.997758	+0.004135
Device 5	0.995974	0.998418	+0.002445
Device 6	0.995571	0.997868	+0.002297
Device 7	0.995448	0.998488	+0.003041
Device 8	0.994553	0.998139	+0.003586
Device 9	0.992126	0.998044	+0.005918
Device 10	0.998156	0.997839	-0.000317
Device 11	0.999163	0.998874	-0.000289

Table 6.11: Weighted Totals for Scenario 3

Referring back to the normalized values, Table 6.4, Device 7 model had the highest thermal values. Device 1 was in the mid-range of the devices thermally, but excelled in efficiency at ambient temperatures. This coupled with the reduced importance placed on cost led to Device 1's success in this scenario.

6.5 Insights

Demonstrated through the EDC component choice process in Section 6.2.1, normalizing values can be applied to reach an unbiased conclusion. In which after reviewing the normalized values the best device was different than the original conclusion. In the end, the TVS 4 model was the best suited device and should be implemented in a SSCB. Using those same normalized values a decision matrix could be developed.

This decision matrix can be used to aid in the component choice process. In which a device can be chosen that best suits the needs of an application. Those needs do not need to be singular in aspect, but can be multifaceted and vary with importance. Even still, the most desired criteria does not need to be known and the use of suppressing the least desired can reach a reasonable solution.

In this thesis an area of focus was power transmission or distribution, which fit closely to Scenario 1, in Section 6.4.1. That case led to the solid-state switch model Device 3 being the best suited for the application. A more economical secondary choice would be Device 11. In review of the normalized value table and the data gathered, this is a reasonable conclusion and Device 3 should be further pursued as a solution for high efficiency SSCBs.

6.6 Recommendations and Future Work

Recommendations for continuing this work are in the area of design and simulation. An area of design that would be useful to explore is removing the redundant gate circuit in a bidirectional SSCB. This would mean removing one of the independent voltage sources and external gate resistors [71]. This may improve real world SSCB control by eliminating a possibility of timing mismatch.

One clear improvement that can be made is the estimated junction temperature curve versus on-state resistance for the United Silicon Carbide models; (TVS 5, TVS 6, TVS 7). A single curve was used for all three models and that curve was estimated from the manufacturer's data sheet. Time could be spent creating a better fit curve or contacting the manufacturer directly for the curve data.

Another aspect of simulation improvement can be made in the use of LTspice simulation options. The options used were to ensure the a timely completion of each of the simulations. Also, the same options were used when simulating each model. Custom fitting the options for each model may be beneficial as well as allowing more time for simulation completion.

Future work in simulation optimization can be achieved with few modifications. The framework for running simulations in LTspice from Matlab with automatic data collection is already completed. As well as, having all of the circuit values be scripted variables in Matlab. This means that every component value or option, be it the external gate resistance or the number of source steps allowed, can be easily changed or cycled to find the optimal value.

CHAPTER 7
EXPERIMENTAL SETUP AND TEST

The following experimental test fixture design would allow the previously simulated SSCB circuits to be fully realized allowing for confirmation of primary components. This test fixture could be applied to a bidirectional or unidirectional SSCB circuit. Any limiting parameters or necessary testing conditions will be explained as needed.

7.1 Experimental Format

The experimental test fixture design will be segregated into two sections encompassing the test fixture design and simulation. The test fixture design will be further divided in to subsections covering the analysis and methodology of each section of the test fixture, as well as implementation. The simulation section will realize the implementation of each test fixture portion, producing a simulated transient response.

7.2 Test Fixture

The test fixture itself can be divided into five main sections: charge circuit, capacitor bank, discharge circuit, device under test (DUT), and load. Each one of the sections has specific design considerations that will be covered in their respective subsections. A block diagram of the test fixture connections and layout can be seen in Figure 7.1.

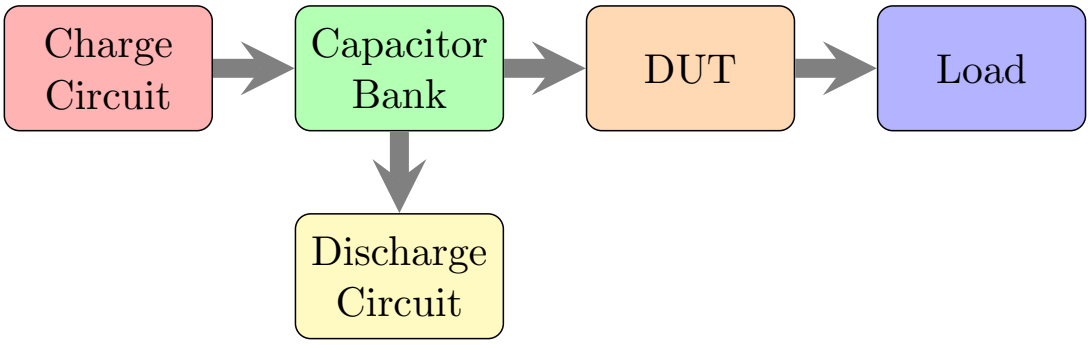


Figure 7.1: SSCB Short Circuit Test Fixture Block Diagram

In Figure 7.1, it can be seen that the charge circuit will supply the capacitor bank. This capacitor bank will have enough energy to complete a single short circuit test. The charge circuit will then be galvanically isolated from the test fixture prior to testing. The capacitor bank will provide all voltage and current to the device under test. The amount of current provided will be determined by the load demand. Finally, the discharge circuit will be used to remove any remaining energy in the capacitor bank after testing. The discharge circuit, device under test, and load will be isolated from each other with a remote switching device.

7.2.1 Charging Circuit

The charging circuit has four main components: voltage source, charge resistor, blocking diode, and galvanic isolation switches. The orientation of these components can be seen in a partial schematic of the test fixture, Figure 7.2.

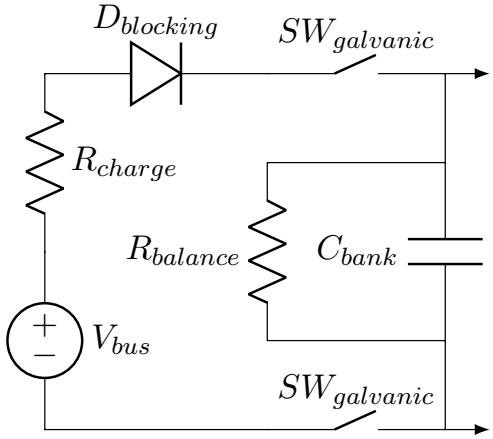


Figure 7.2: Charging Circuit Portion of SSCB Short Circuit Test Fixture

The voltage source can be any power supply that has a maximum voltage level at or above the desired testing voltage. Specifically, the voltage source can have any current supply limit, but must be able to provide enough potential to charge the capacitors. The current supply limit only affects the charging time of the capacitor bank. In short, a high voltage low current supply will meet the needs of this thesis, in addition power supplies of this makeup tend to be more economical.

The charge resistor also affects the charging time of the capacitor bank as well as the inrush current. This component may not be necessary if the voltage supply being used has a low enough current supply limit. Most commonly when designing a charge circuit for capacitors, the capacitor ripple current limit is the most limiting factor. In this case, DC voltage is being supplied, therefore no ripple is present and a charge resistor is unnecessary.

The blocking diode will prevent any current flow from the capacitor bank to the voltage source. Internal to most power sources exist a discharge circuit. This circuit if left connected without a blocking diode would force the galvanic switch to break the voltage of the capacitor bank. Due to the installation of this blocking diode a lower rated switch can be used as long as a strict order of operations is maintained. As in, the galvanic switch is only operated when the voltage source is not operating.

A galvanic isolation switch will isolate the power supply from the test fixture and from the text fixture ground as well. This type of isolation is necessary due to the high energy discharge during testing. This type of discharge may trip safety devices within the power supply or damage the power supply itself.

7.2.2 Implementation

The variables previously discussed in this section and their values, either derived by design or determined by manufacturer are displayed in Table 7.1. The specific components chosen to fulfill the above design parameters in Table 7.1 and displayed with their relevant values in Table 7.2.

Variable	**Units**	**Value**
V_{bus}	(V)	1000
R_{charge}	(Ω)	0
$D_{blocking}$	(V)	1000
$SW_{galvanic}$	(mA)	20

Table 7.1: Charge Circuit Derived Values for the SSCB Shorting Test Fixture

	$\mathbf{V}_{bus}$	$\mathbf{R}_{charge}$	$\mathbf{D}_{blocking}$	$\mathbf{SW}_{galvanic}$
Manufacturer	Power Designs	-	IXYS	Altech Corp.
Model	1544	-	UGE1112AY4	KEM340UL Y/R
V_{rated} (V)	3012	-	8000	600
I_{rated} (A)	$20m$	-	2	40
R_{rated} (Ω)	-	-	-	-

Table 7.2: Charge Circuit Components of the SSCB Shorting Test Fixture

7.2.3 Capacitor Bank

The capacitor bank has two main components: a capacitor network and a balancing resistor network as seen in the previous charging circuit diagram, Figure 7.2. The capacitor bank is fully isolated from the power supply by the galvanic switches during testing. Further still, the capacitor bank is isolated from the DUT by a remote switch. The remote switch does not need to provide galvanic isolation and can be a solid-state device. Though, the remote switch must be rated for the transient current. This same remote switch will be used elsewhere in the test fixture including the discharge circuit and load, and will be covered in detail in Section 7.2.9.2.

7.2.3.1 Capacitance Value

Choosing a specific capacitor or capacitors requires knowing the desired capacitance value of the entire bank. This can be determined from the capacitor discharge equation, Equation 7.1 [3]. In which the final voltage of a capacitor after a discharge event can be determined from the initial voltage of the capacitor, resistance in the discharge path, total capacitance value of the capacitor, and the total discharge time of the event.

$$V_c = V_o \, e^{\frac{-t}{R\,C}} \tag{7.1}$$

In this case we desire to know the capacitance value (C) and can rearrange 7.1 to suit our needs, as seen in Equation 7.2. Therefore, we must determine the resistance in the discharge path (R), initial capacitor voltage (V_o), the final capacitor voltage (V_c), and

the desired time to transition between those voltage levels (t).

$$C = \frac{-t}{R\, ln(V_c/V_o)} \tag{7.2}$$

The resistance in the discharge path of the capacitor bank can be estimated as being the same as the shorting load value. A more precise value can be determined using the resistance of the DUT, remote switches, cabling as well as the steady state resistive load in parallel with the shorting load. This determination can be seen in Equation 7.3.

$$R_{max} = R_{SW_1} + R_{cable} + R_{DUT} + R_{cable} + \left(\frac{1}{(R_{load} + R_{SW_4})} + \frac{1}{(R_{short} + R_{SW_3})} \right)^{-1} \tag{7.3}$$

The initial voltage of the capacitor bank is the testing voltage for the DUT. In other words, the desired potential to be experienced by the DUT, determines the voltage charge on the capacitor bank. This can also be considered to be the test fixture bus voltage. This value is arbitrary and determined by the designer.

The final voltage of the capacitor bank is the desired voltage level to be reached after a discharge event. This loss in voltage is referred to as voltage droop and is normally measured in percent dropped, as seen in Equation 7.4. A voltage source or bus voltage that has the ability to maintain its potential after a discharge event would have a small voltage droop and be considered a stiff source. An example of a small voltage droop is 1%, but a more reasonable value would be 5% or even 10% drop in bus voltage. As with the initial voltage, the final capacitor voltage is an arbitrary value and determined by the designer.

$$V_{final} = V_{bus} - (V_{bus} \times \%_{droop}) \tag{7.4}$$

The desired time to transition between initial and final capacitor bank voltages is the discharge time. This time value, like the voltage values, is arbitrarily defined by the designer. A normal industry expectation for a circuit breaker to clear any fault is approximately $500ms$. Taking care to ensure enough energy is being supplied throughout this time duration would more than suffice most testing purposes. As noted in Chapter

2.5 Literature Review, SSCBs response time is in the micro-seconds and significantly shorter discharge time can be adopted.

Equation 7.2 can be redefined as Equation 7.5, in which the following variables have been defined: the desired capacitance value ($C_{desired}$), the resistance in the discharge path (R_{max}), the initial capacitor bank voltage ($V_{cap,o}$), the final capacitor bank voltage ($V_{cap,f}$), and the desired time to transition between those voltage levels ($t_{discharge}$).

$$C_{desired} = \frac{-t_{discharge}}{R_{max}\, ln(V_{cap,f}/V_{cap,o})} \tag{7.5}$$

7.2.3.2 Capacitor Selection

Once a total capacitor value has been determined, specific components can be chosen. In most cases a single capacitor will not meet the application requirements. Therefore a capacitor network will need to be designed. This network could include capacitors in series as well as in parallel and could be thought of as a matrix of capacitors, as seen in Figure 7.3. Where the number of capacitors in series is equivalent to the rows, (m_{cap}), and the number of parallel branches is equivalent to the columns, (n_{cap}).

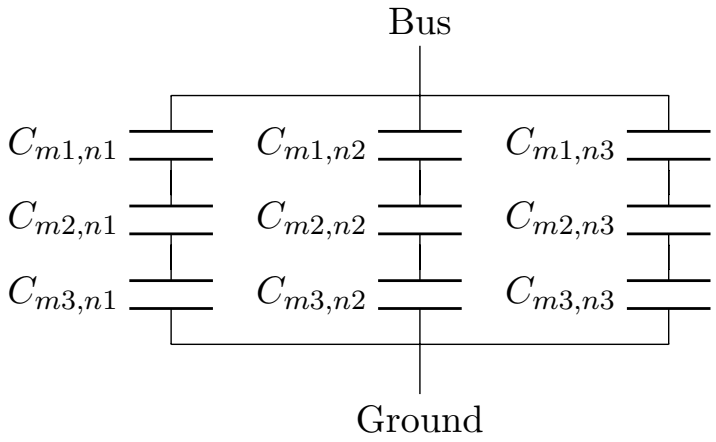


Figure 7.3: Capacitor Bank Matrix

The number of capacitors placed in series is determined by the desired testing voltage and the rated operating voltage of the capacitor. In that a single capacitor may

not be rated to withstand the full bus voltage. Therefore the number of required capacitors is determined by dividing the initial voltage of the capacitor bank by the rated voltage of an individual capacitor. This determination can be seen in Equation 7.6, in which the result would be rounded up to the next whole number.

$$m_{cap} = \frac{V_{cap,o}}{V_{rated}} \tag{7.6}$$

Placing capacitors in series reduces the effective capacitance. Therefore, parallel branches must be added to increase the capacitance to the desired level. Each parallel branch is dependent on this desired capacitance level, the individual capacitor value, and of course the number of capacitors in series. This determination can be seen in Equation 7.7. Much the same as with the series number of capacitors, the result of the parallel branches must be rounded up to the next whole number. This may result in a capacitance value that does not match exactly as desired.

$$n_{cap} = \frac{C_{desired}}{(C_{single}/m_{cap})} \tag{7.7}$$

7.2.3.2.1 Capacitor Comparison

The capacitor selection process has many variables that can end with varying results, but all are dependent on the application requirements. As previously discussed, the number of individual capacitors in the capacitor bank will vary with model. If the desired total capacitance level has a tight tolerance, then smaller individual capacitors would be better suited. The smaller increments of capacitance would afford a higher fidelity in control, though achieving an exact number is unlikely.

Having a large number of capacitors introduces it's own set of variables; including individual capacitor tolerances and cost. Individual capacitors have manufactured tolerances and will cause variances even within the same component model family. This variance can create a situation in which an individual capacitor experiences and over voltage condition. Understanding the severity of a potential over voltage condition can

help guide component choice. This can be accomplished using Equation 7.8 in which the series number of capacitors is greater than one and the tolerance value is set by the manufacturer; which normally is 20%.

$$V_{variance} = V_{cap,o} \times \frac{(100\% + \%_{tolerance})}{(100\% + \%_{tolerance}) + (m_{cap} - 1) \times (100\% - \%_{tolerance})} \tag{7.8}$$

Another aspect of component choice is price. Much like the previous analysis with solid-state switches, this can be expressed in price per device ($Price_{unit}$), total price ($Price_{total}$), or price per energy density ($Price_{energy}$). Determining the total price can be accomplished using Equation 7.9, which is dependent on the single device price and the earlier defined rows and columns of the capacitor matrix.

$$Price_{total} = Price_{unit} \times m_{cap} \times n_{cap} \tag{7.9}$$

Determining the price per energy density can be accomplished using Equation 7.10, which is dependent on the single device price and the energy density of the device. The energy density can be calculated using the individual capacitor value and the individual capacitor voltage rating, as seen in Equation 7.11 [3].

$$Price_{energy} = \frac{Price_{unit}}{w_{capacitor}} \tag{7.10}$$

$$w_{capacitor} = \frac{1}{2} \times C_{single} \times V_{rated}^2 \tag{7.11}$$

7.2.3.3 Balancing Network

Once a suitable capacitor has been chosen an accompanying balancing network must be designed. This balancing network will ensure even charging and discharging of the capacitor bank. Maintaining an even charge on the capacitor bank will promote component longevity and reduce the chances of failures. The individual resistors in the network are installed in parallel to the individual capacitors, as seen in Figure 7.4.

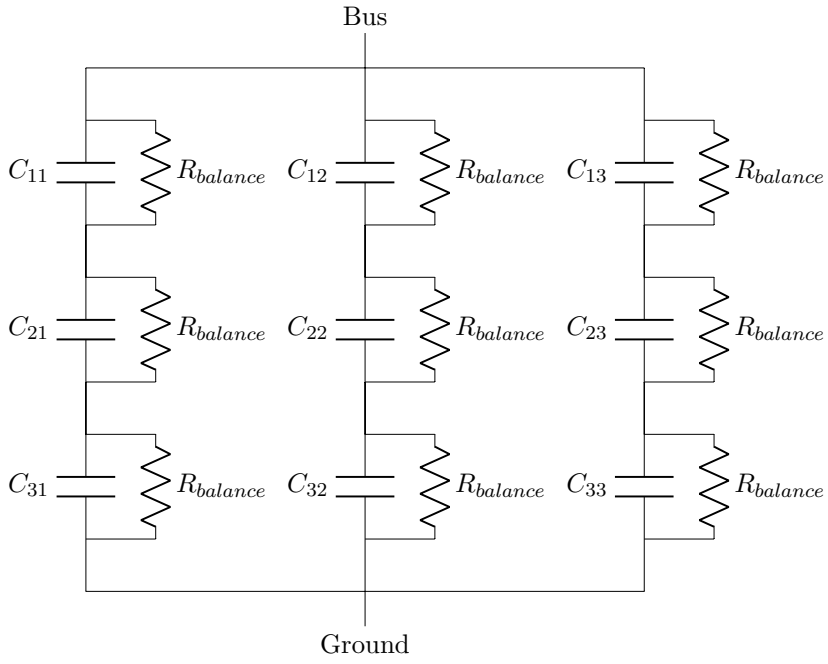


Figure 7.4: Resistor Network Matrix

There are two main considerations when designing a resistor balancing network. First, the resistance value of each resistor ($R_{balance}$). Second, the expected power dissipation of each resistor, ($P_{R,balance}$). The resistance value is dependent on the number of capacitors in series, the rated voltage of each capacitor, the leakage current of each capacitor, and the applied voltage to the entire capacitor bank or bus voltage. These dependencies can be seen in Equation 7.12.

$$R_{balance} = \frac{(m_{cap} \times V_{rated} - V_{cap,o})}{I_{leakage}} \tag{7.12}$$

The most leakage current values are determined by the manufacturer for specific models and capacitance values. For instance, KEMET's ALS70/71 series of capacitors the leakage current is defined for two groups of capacitance. If rated capacitance is $<$ 330,000 μF, then leakage current is equal to Equation 7.13 or 6,000 μA, whichever is smaller. If rated capacitance is $\geq$ 330,000 μF, then leakage current is 16,000 μA.

$$I_{leakage}(\mu A) = 0.006 \times C_{rated}(\mu F) \times V_{rated}(VDC) \tag{7.13}$$

The expected power dissipated by each resistor can be determined using the previously determined resistance value from Equation 7.12 as well as the previously derived leakage current. This determination can be seen in Equation 7.14.

$$P_{R,balance} = I^2_{leakage} \times R_{balance} \tag{7.14}$$

7.2.4 Implementation

The variables previously discussed in this section and their values, either derived by design or determined by manufacturer are displayed in Table 7.3. The specific components chosen to fulfill the above design parameters in Table 7.3 and displayed with their relevant values in Table 7.4.

Variable	**Units**	**Value**
$V_{cap,o}$	(V)	1000
$\%_{droop}$	$(\%)$	10
$V_{cap,f}$	(V)	900
R_{max}	(Ω)	6.35819
R_{cable}	(Ω)	0.3216
R_{DUT}	$(m\Omega)$	160
$R_{SW1,2,3,4}$	(Ω)	1.566
R_{load}	(Ω)	30
R_{short}	(Ω)	3
$t_{discharge}$	(ms)	1
$C_{desired}$	(mf)	1.49276
m_{cap}	-	3
n_{cap}	-	2
$V_{variance}$	(V)	250
$\%_{tolerance}$	$(\%)$	20
$Price_{total}$	($)	804.34
$Price_{unit}$	($)	133.89
$Price_{energy}$	$(\$/J)$	0.0823938
$w_{capacitor}$	(J)	1625
C_{single}	(μF)	13000
V_{rated}	(V)	500
$R_{balance}$	$(k\Omega)$	100
$I_{leakage}$	(μA)	6000
$P_{R,balance}$	(W)	3.6

Table 7.3: Capacitor Bank Derived Values of the SSCB Shorting Test Fixture

	$\mathbf{C}_{bank}$	$\mathbf{R}_{balance}$
Manufacturer	Kemet	TE
Model	ALS70A133QT500	3550 Series
V_{rated} (V)	500	300
I_{rated} (A)	26.5	-
R_{rated} (Ω)	23	$100k$
P_{rated} (W)	-	5
C_{rated} (F)	$13m$	-
C_{bank} (F)	$8.6667m$	-

Table 7.4: Capacitor Bank Components for the SSCB Shorting Test Fixture

Electronic component distributors were surveyed for potential capacitor bank components, resulting in 6751 candidates. Applying the previously described comparison criteria for capacitors will alleviate some of the burden of choosing a specific component. The first step in the down selection process should be to sort the selection pool by $Price_{energy}$, this will reveal the most energy dense components for the price. Next, review the top 25 candidates for other desired criteria such as number of required capacitors, total bank capacitance, or impedance.

The comparison of desired criteria depends on the application. In this case, the capacitor bank should have the ability to withstand $1250V$, expecting a 10% droop, with a desired total capacitance of $1.49276mF$. Within the top candidates the model that most closely fit the desired total bank capacitance was the United Chemi-Con: ESMQ451VSN561MR505, with a total capacitance of $1.49333mF$. Though, this model would require four columns of eight capacitors for a total of twenty four. This number of capacitors is undesirable. A second candidate with a more reasonable total was the Kemet: ALS70A752NJ400. This model would only require one column of four capacitors, with a total capacitance of $1.875mF$.

In the end the capacitor model that was chosen allows for the experimental test fixture to take full advantage of the chosen power supply capabilities. This means the desired withstand voltage was changed to $3000V$. With this change, the Kemet: ALS70A133QT500 became the desired model, requiring only a single column of six capacitors to meet this threshold. The configuration of the capacitor bank can be changed to meet the needs of the application. In this instance the single column of six capacitors would be reconfigured to be two columns of three capacitors each.

7.2.5 Discharging Circuit

Discharge circuits help facilitate the removal of stored energy in the system, in a timely manner. A test fixture of this capacitance level without a discharge circuit would take a considerable amount of time to reach a safe voltage level. When designed, installed, and used properly a discharge circuit is key to personnel safety. The discharge circuit has two main components being a resistive load and remote switch, as displayed in Figure 7.5.

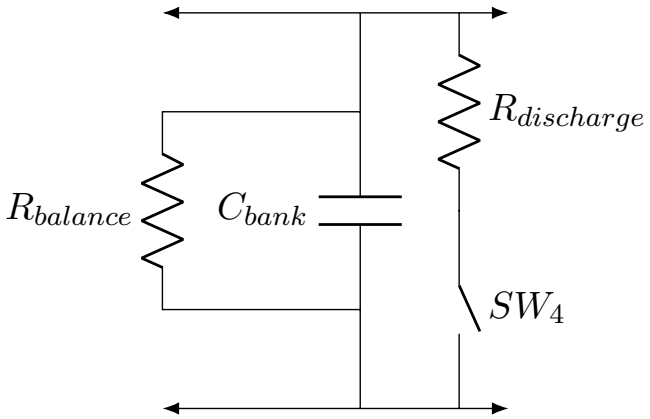


Figure 7.5: Discharging Circuit Portion of SSCB Short Circuit Test Fixture

The value of the resistive load can be determined using the capacitor discharge equation, as seen in Equation 7.1, but rearranged to solve for the resistance value. In Equation 7.15, the capacitor voltage would be considered to be initial capacitor bank voltage and the final voltage would be near zero. The discharge time is arbitrarily defined by the designer, but should account for stress and heating of the capacitor bank.

$$R_{discharge} = \frac{-t_{discharge}}{C_{bank} \times ln(V_{cap,f}/V_{cap,o})} \tag{7.15}$$

Much like the balancing resistors, another aspect of the resistive load that must be taken into account is the expected power dissipation of the discharge resistor or resistors. Determining the expected power dissipation can be accomplished using a simple power equation, as seen in Equation 7.16. This power equation is similar to the balancing resistor power equation, but the bus voltage is not divided by the number of capacitors.

This is because the entire bus potential will be felt by the discharge circuit.

$$P_{R,discharge} = \frac{(V_{cap,o})^2}{R_{discharge}} \tag{7.16}$$

The final component of the discharge circuit is the remote switch. As previously mentioned, the remote switch will be used elsewhere in the test fixture and will be covered in detail in Section 7.2.9.2.

7.2.6 Implementation

The variables previously discussed in this section and their values, either derived by design or determined by manufacturer are displayed in Table 7.5. The specific components chosen to fulfill the above design parameters in Table 7.5 and displayed with their relevant values in Table 7.6.

Variable	**Units**	**Value**
$V_{cap,o}$	(V)	1000
$V_{cap,f}$	(V)	900
$R_{discharge}$	(Ω)	5011.09
$t_{discharge}$	(s)	100
C_{bank}	(mf)	8.6667
$P_{R,discharge}$	(W)	200

Table 7.5: Discharge Circuit Derived Values of the SSCB Shorting Test Fixture

	$\mathbf{R}_{discharge}$	$\mathbf{SW}_4$
Manufacturer	TE	IXYS
Model	E12 Series	IXBF50N360
V_{rated} (V)	2500	3600
I_{rated} (A)	-	420
R_{rated} (Ω)	$10k$	1.566
P_{rated} (W)	100	-

Table 7.6: Discharge Circuit Components for the SSCB Shorting Test Fixture

7.2.7 Device Under Test

The purpose of this experimental test fixture is to supply enough voltage and current to a device under test (DUT) in order to verify the performance characteristics. In the DUT portion of this test fixture there are three main components being interconnecting cabling, the DUT, and a remote switch, as displayed in Figure 7.6.

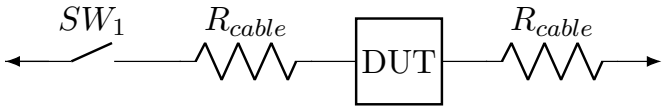


Figure 7.6: Device Under Test Portion of SSCB Short Circuit Test Fixture

In the case of this thesis the DUT would be a SSCB of either unidirectional or bidirectional configuration. Though any circuit breaker type or design can be tested in this experimental test fixture. The DUT is connected to the capacitor bank via interconnecting cabling through a remote switch. Similarly, the DUT is connected to the load via interconnecting cabling, though a remote switch is not necessary in this portion of the test fixture.

The interconnecting cabling can be of almost any configuration, as long as it meets the current carrying requirements of the test. The maximum current of the test, which is the same as the current during a shorting transient, can be determined by Ohm's law. An estimate can be determined using the shorting load or a more precise value can be calculated using the series resistance, as seen in Equation 7.17.

$$I_{max} = \frac{V_{cap,o}}{R_{max}} = I_{short} \tag{7.17}$$

Once the amount of current is determined an appropriate current carrying conductor can be determined. This conductor will have an inherent resistance. Accounting for this resistance in the discharge path will determine a more accurate capacitance value, as seen in Equation 7.3. An industry standard interconnecting cable,

often used in high current applications is $4/0AWG$ cabling. This size of cabling has an ampacity of $206A$ at $90°C$ or less per conductor. As well as an estimated resistance of $0.1608m\Omega$ for every $1m$ in length.

7.2.8 Implementation

The variables previously discussed in this section and their values, either derived by design or determined by manufacturer are displayed in Table 7.7. The specific components chosen to fulfill the above design parameters in Table 7.7 and displayed with their relevant values in Table 7.8.

Variable	**Units**	**Value**
$V_{cap,o}$	(V)	1000
I_{max}	(A)	157.277
R_{max}	(Ω)	6.35819

Table 7.7: DUT Portion Derived Values of the SSCB Shorting Test Fixture

	DUT	$\mathbf{R}_{cable}$	$\mathbf{SW}_1$
Manufacturer	Microchip Technology	-	IXYS
Model	MSCO80SMA120J	4/0	IXBF50N360
V_{rated} (V)	1200	-	3600
I_{rated} (A)	37	206	420
R_{rated} (Ω)	$160m$	0.3216	1.566
Length (m)	-	2	-

Table 7.8: DUT Portion Components for the SSCB Shorting Test Fixture

It is worth noting that the EDC component chosen to be simulated was the Littelfuse Inc.: AK3-680C. This component choice as well as the EDC as a whole does not impact any design decisions of the experimental test fixture. As a reminder, the Littelfuse Inc.: AK3-680C had the highest normalized value score of the selection pool.

7.2.9 Load

The load circuit determines the steady state current as well as the shorting current. In this portion of the test fixture there are four main components: the resistive load, the shorting load, remote switching devices, and the inductive load, as seen in Figure 7.7.

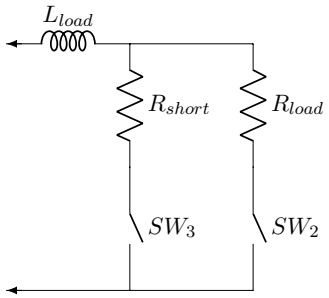


Figure 7.7: Load Portion of SSCB Short Circuit Test Fixture

7.2.9.1 Resistive Loads

The resistive load values determine the steady state load current and the shorting transient current. The steady state full load value is dependent on the DUT current capabilities. In this thesis full load is considered to be 30Ω, as seen in previous chapters. This full load can be used to estimated the steady state current, but a more precise value can be determined using Equation 7.18. In which the resistances of all the components in the current path are taken into account, as seen in Equation 7.19.

$$I_{steady\ state} = \frac{V_{cap,o}}{R_{steady\ state}} \tag{7.18}$$

$$R_{steady\ state} = R_{SW,1} + R_{cable} + R_{DUT} + R_{cable} + R_{load} + R_{SW,2} \tag{7.19}$$

The desired shorting current limit is dependent on the EDC or vice versa. An earlier determination of the shorting load value was made in Chapter 3: Analysis and Design Section 3.4.3.1. In which Equation 3.28 was used to determine the shorting load value for this thesis to be 3Ω. This values was used to determine the maximum current during a shorting transient as seen in Equation 7.17.

7.2.9.2 Remote Switches

The two main aspects of using remote switching devices are safety and control. As for safety, using remote switching devices allows lab personnel the ability to operate the test fixture from a distance outside any arc flash boundary. Also, remote operation allows

for the automation of the testing sequence and data collection. In particular the coordiation of test equipment, such as oscilloscopes to capture the shorting transient.

7.2.9.3 Inductive Load

Choosing a specific inductor starts with knowing the desired inductance value. This can be determined from the inductor charge equation, Equation 7.20. In which the final current of an inductor after a charge event can be determined from the initial current of the inductor, resistance in the charge path, total inductance value of the inductor, and the total charge time of the event.

$$I_l = I_o \left(1 - e^{\frac{-t}{L/R}}\right) \tag{7.20}$$

In this case we desire to know the inductance value (L) and can rearrange 7.20 to suit our needs, as seen in Equation 7.21. Therefore, we must determine the resistance in the charge path (R), initial inductor current (I_o), the final inductor current (I_l), and the desired time to transition between those current values (t).

$$L = \frac{-t \times R}{ln\left(1 - \frac{I_l}{I_o}\right)} \tag{7.21}$$

The initial inductive load current can be considered the same as the shorting current, as determined from Equation 7.17. The final inductive load current is similar to the shorting current but is calculated using the final voltage after the desired droop, as determined using Equation 7.22. The series resistance was earlier defined in Equation 7.3 and has not changed. Much like the discharge time of the capacitor bank, the charge time of the inductive load is arbitrarily defined by the designer.

$$I_{droop} = \frac{V_{cap,o} - (V_{cap,o} \times \%_{droop})}{R_{max}} \tag{7.22}$$

Equation 7.21 can be redefined as Equation 7.23, in which the following variables have been defined: the desired inductance value ($L_{desired}$), the resistance in the charge

path (R_{max}), the initial inductive load current ($I_{steady\ state}$), the final inductive load current (I_{max}), and the desired time to transition between those current values (t_{charge}).

$$L_{desired} = \frac{-t_{charge} \times R_{max}}{ln\left(1 - \frac{I_{droop}}{I_{max}}\right)} \tag{7.23}$$

Once the inductive load value is determined an inductor can be selected from an electronic component distributor. More than likely the exact desired value will not be available. An alternative to purchasing a component is to design and make a custom component. The equation for inductance in an air coil, Equation 7.24, can be used to determine the design parameters for a custom inductor [3].

$$L_{desired} = \mu_r\, \mu_o\, \frac{(turns^2\, area)}{length} \tag{7.24}$$

The permeability of free space (μ_o) is a known value of $0.4\pi \times 10^{-8}$, as well as the permeability of core (μ_r), which is air and has a value of 1. The area of the inductor must be arbitrarily chosen by the designer. Keep in mind implementation of the design when deciding this value and chose an appropriate value. The flexibility of the cabling used could limit the diameter of the desired inductor. The same cabling used for the interconnection of the device under test could be re-appropriated for the inductor. The length of the inductor can be determined by the diameter of the cabling used and the number of turns. Therefore the length variable in Equation 7.24 should be substituted accordingly. Once these values are determined, the Equation 7.24 can be rearranged to solve for the number of turns in the inductor, as seen in Equation 7.25 [3].

$$turns = \frac{L_{desired} \times D_{cable}}{\mu_r \times \mu_o \times \pi \left(\frac{D_{inductor}}{2}\right)^2} \tag{7.25}$$

The final turns value should be rounded up to the nearest whole number, this will facilitate implementation. Then, a new inductor value should be calculated using the new rounded turns value and Equation 7.24. This final inductor value will be used for the inductive load value.

7.2.10 Implementation

The variables previously discussed in this section and their values, either derived by design or determined by manufacturer are displayed in Table 7.9. The specific components chosen to fulfill the above design parameters in Table 7.9 and displayed with their relevant values in Table 7.10.

Variable	**Units**	**Value**
$V_{cap,o}$	(V)	1000
$I_{stead\ state}$	(A)	29.4679
$R_{steady\ state}$	(Ω)	33.9352
R_{cable}	(Ω)	0.3216
R_{DUT}	$(m\Omega)$	160
$R_{SW1,2}$	(Ω)	1.566
R_{load}	(Ω)	30
$L_{desired}$	(μH)	138.064
R_{max}	(Ω)	6.35819
t_{charge}	(μs)	50
I_{max}	(A)	157.277
$\%_{droop}$	$(\%)$	10
I_{droop}	(A)	141.55
turns	-	11
D_{cable}	(mm)	19.3
$D_{inductor}$	(m)	0.5
μr	-	1
μo	-	$0.4\pi \times 10^{-8}$

Table 7.9: Load Portion Derived Values of the SSCB Shorting Test Fixture

	$\mathbf{L}_{load}$	$\mathbf{R}_{load}$	$\mathbf{R}_{short}$	$\mathbf{SW}_{3,4}$
Manufacturer	-	TE	TE	IXYS
Model	-	E12 Series	E12 Series	IXBF50N360
$V_{rated}\ (V)$	-	2500	2500	3600
$I_{rated}\ (A)$	-	-	-	420
$R_{rated}\ (\Omega)$	-	30	3	1.566
$P_{rated}\ (W)$	-	50	50	-
$L_{rated}\ (H)$	140.63μ	-	-	-

Table 7.10: Load Portion Components for the SSCB Shorting Test Fixture

7.2.11 Final Design

As previously determined, the test fixture has five main sections: charge circuit, capacitor bank, discharge circuit, device under test, and load. Using the above design methods each one of the section can be more specifically defined as seen in Figure 7.8.

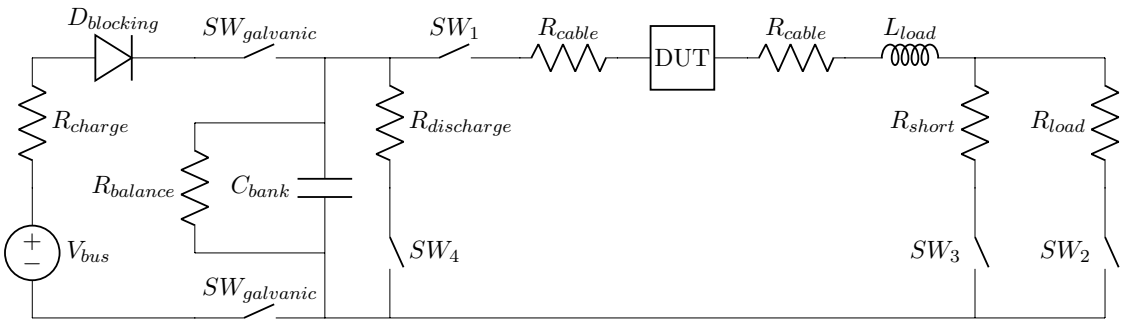


Figure 7.8: SSCB Short Circuit Test Fixture

7.2.12 Charging

A charging sequence of the capacitor bank has been completed using the simulated circuit seen in Figure 7.9. This charging sequence has a duration of $526s$ or $8m$ and $46s$ having been initiated at $0V$ and ending at $1000V$.

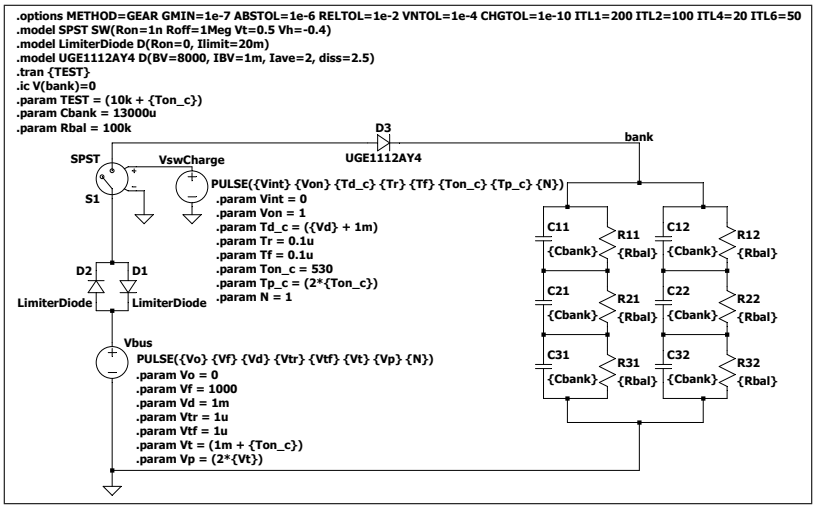


Figure 7.9: SSCB Short Circuit Test Fixture Charging Sequence Simulation

Allowing the capacitor bank to discharge to a safe voltage level without intervention would take an unreasonable amount of time. As simulated, after more than $4,000s$ or over an hour the capacitor bank reached low voltage, $< 50V$. This behavior can be seen in Figure 7.10.

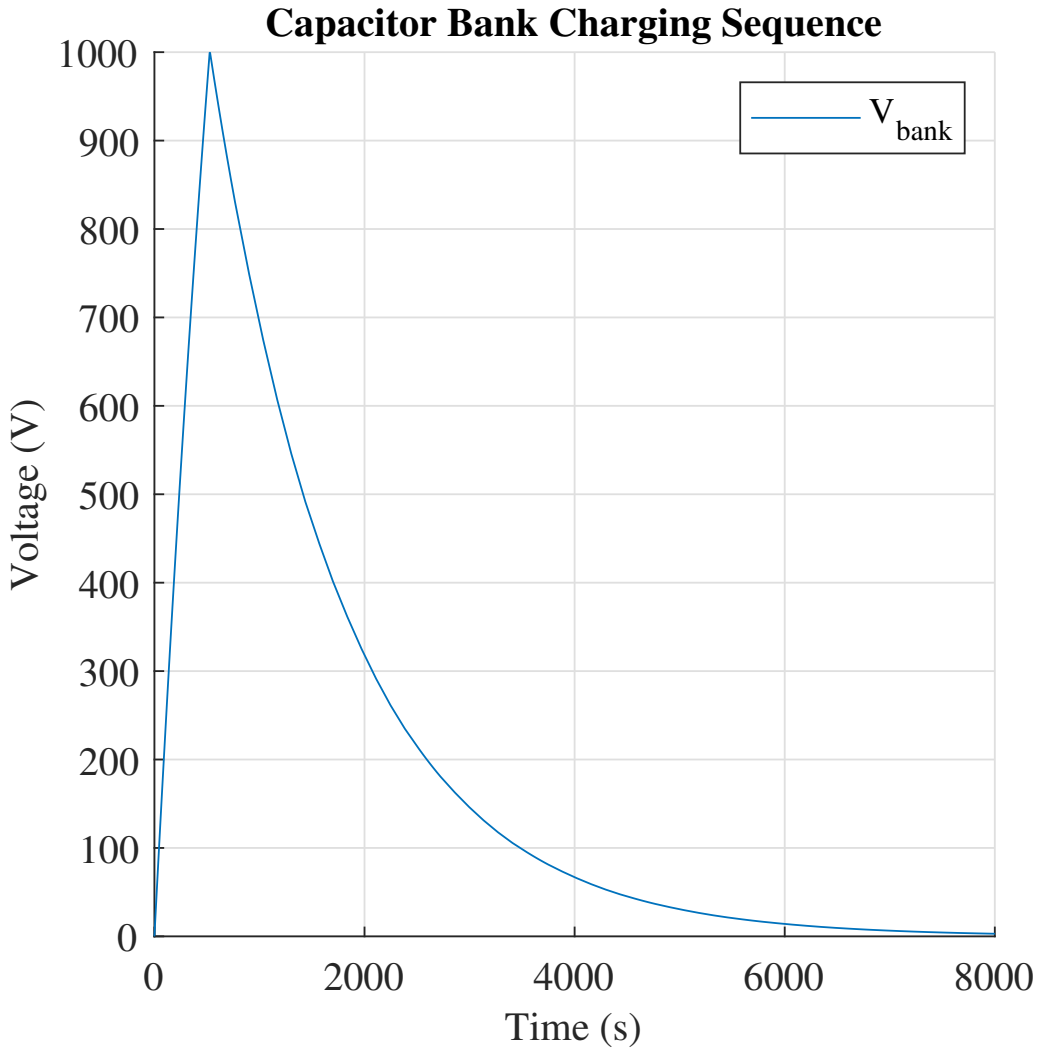


Figure 7.10: SSCB Short Circuit Test Fixture Charging Sequence Behavioral Waveform

This result reiterates the fact a discharge circuit is necessary in this test fixture and that having a discharge circuit installed will ensure safe and timely operation of the test fixture.

7.2.12.1 Discharging

A discharging sequence of the capacitor bank has been completed using the simulated circuit seen in Figure 7.11. The discharging sequence has a duration of $289s$ or $4m$ and $49s$ having been initiated at $1000V$ and ending with less than $1V$. The behavior of the discharging sequence can be seen in Figure 7.12.

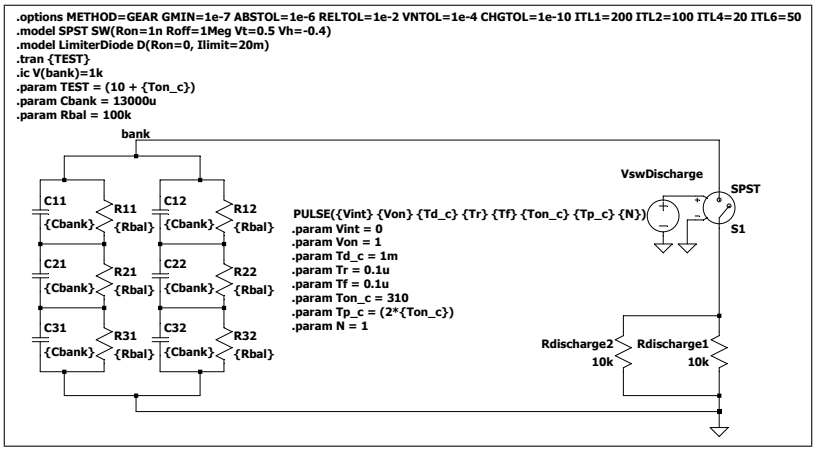


Figure 7.11: SSCB Short Circuit Test Fixture Discharging Sequence Simulation

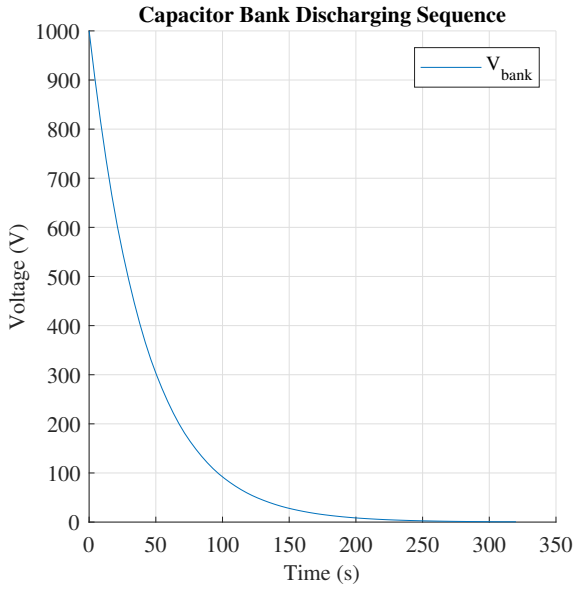


Figure 7.12: SSCB Short Circuit Test Fixture Discharging Sequence Behavioral Waveform

7.2.12.2 Recharging

The experimental test fixture may not always be fully discharged between tests. Estimating how long a potential recharging sequence has been completed using the simulated circuit seen in Figure 7.13. The recharging sequence was initiated after a charging sequence and shorting transient. The charging sequence was initiated at $0V$ with the shorting transient being initiated at $1000V$, lasting for $1ms$. As can be seen in Figure 7.14, the capacitor bank dipped to $954V$ with a recharging duration of $26.4s$.

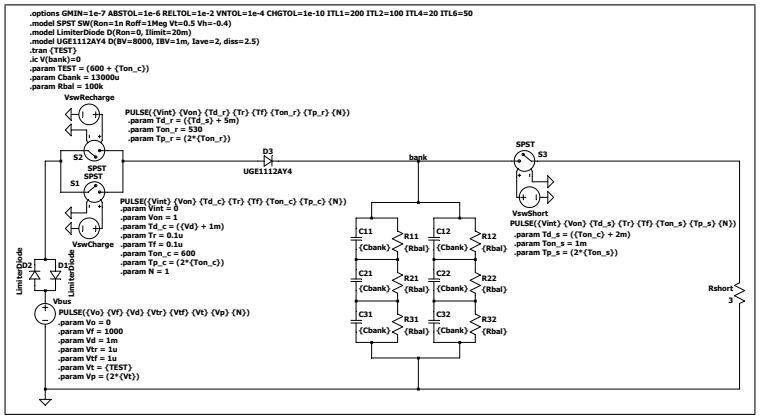


Figure 7.13: SSCB Short Circuit Test Fixture Recharging Sequence Simulation

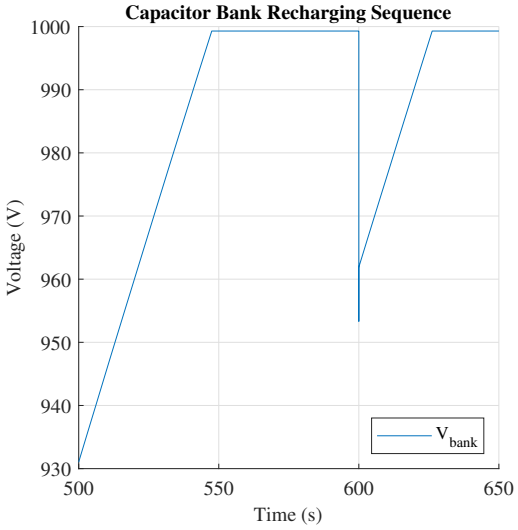


Figure 7.14: SSCB Short Circuit Test Fixture Recharging Sequence Behavioral Waveform

7.2.12.3 Shorting Transient Simulation

A simulated shorting transient sequence was completed using the circuit seen in Figure 7.15. At the start of the simulation the capacitor bank was initially charged to $1000V$. Then a particular timing sequence of switches was enacted. That sequence is as follows: at $1ms$ the load was connected (SW_2), at $2ms$ the DUT was connected (SW_1), at $3ms$ the DUT was operated, and at $4ms$ the shorting load was connected (SW_3). As can be seen in Figure 7.16, the capacitor bank discharged $5V$ during the steady state portion of the transient and ending at approximately $968V$ after the shorting transient.

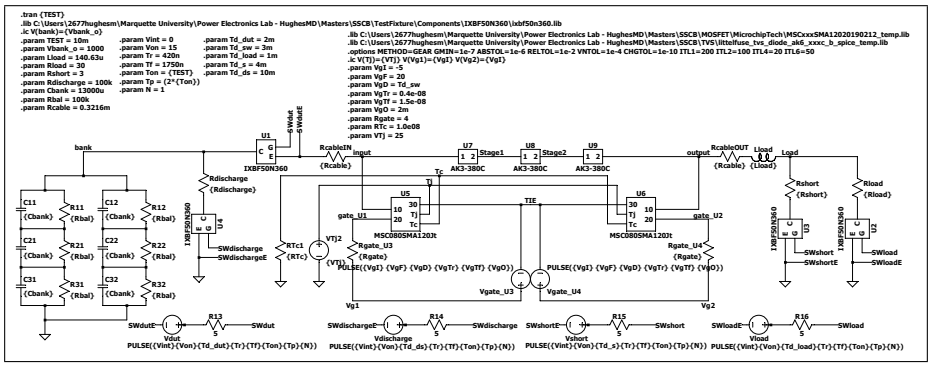


Figure 7.15: SSCB Short Circuit Test Fixture Simulation

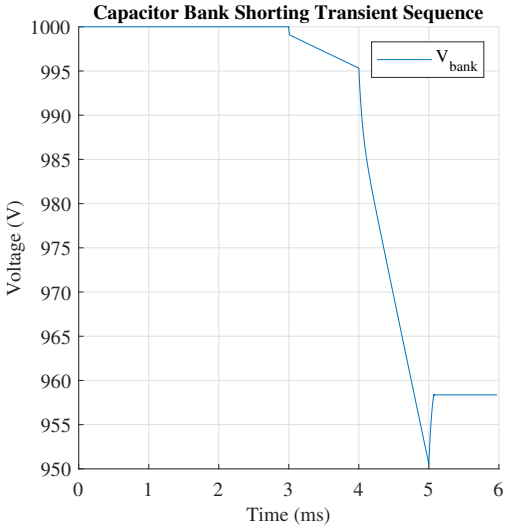


Figure 7.16: SSCB Short Circuit Test Fixture Shorting Transient Sequence

The affect of the shorting transient on the DUT can be seen in Figure 7.17, in which the difference between the input voltage and output voltage across the DUT is displayed with the current experienced by the DUT.

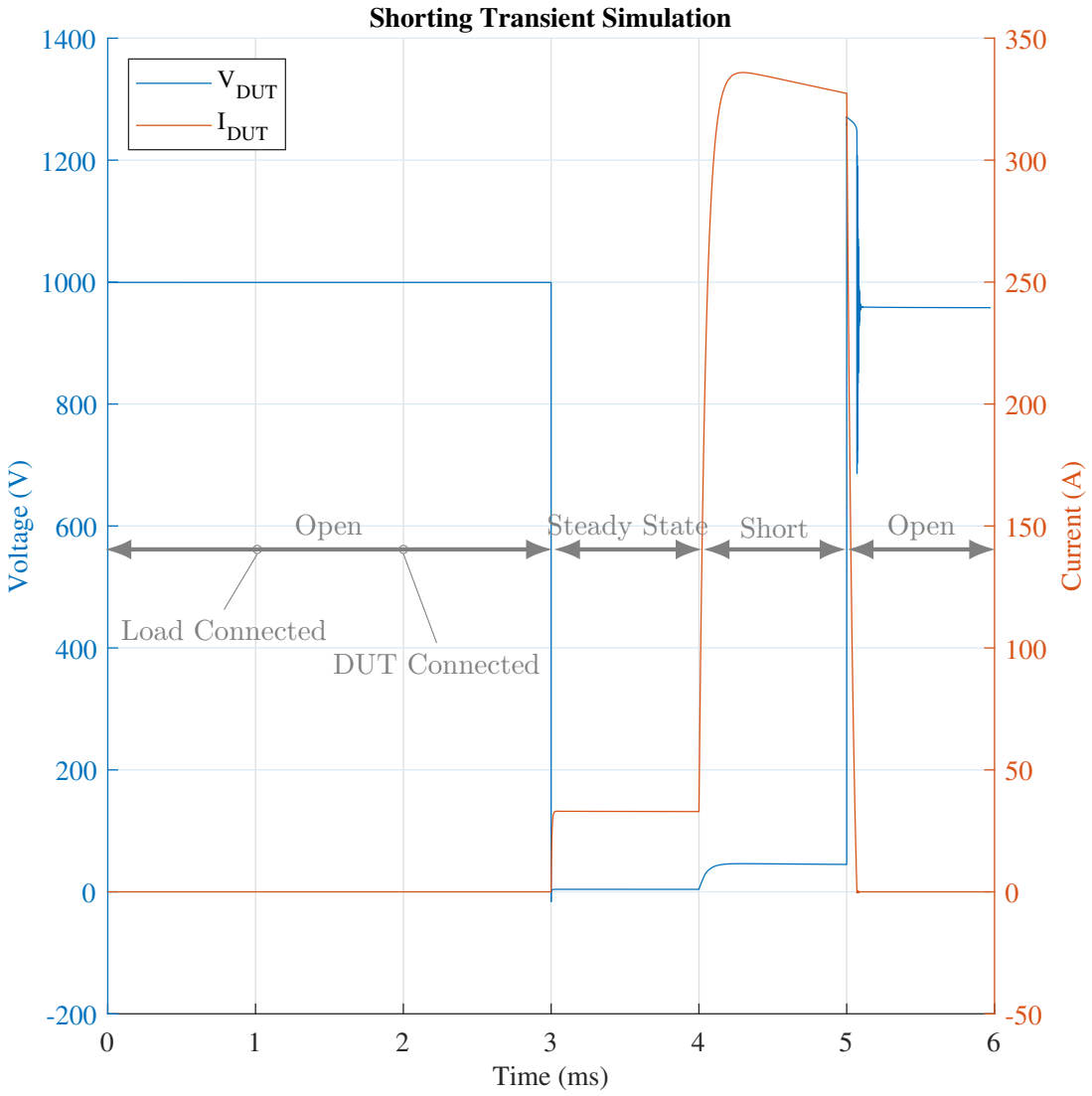


Figure 7.17: SSCB Short Circuit Test Fixture Shorting Transient Behavioral Waveform

The DUT was able to withstand the full bus voltage prior to closing at time $3ms$. Once closed, the voltage across the DUT dropped to near $4V$ with a steady state current of approximately $33A$. Then the shorting transient was initiated at $4ms$ and the voltage drop across the DUT peaked near $46V$ with a current peak of $336A$. During the shorting transient the DUT was opened, causing a voltage protection clamping event to occur.

This protection event can be seen more closely in Figure 7.18, in which the time scale has been reduced to highlight the event. In Figure 7.18 it can be seen that the voltage peaked at $1271V$ with $323A$ of current present. The event was completed in less than $0.1ms$, returning the voltage to $959V$ across the DUT and $0A$ of current present.

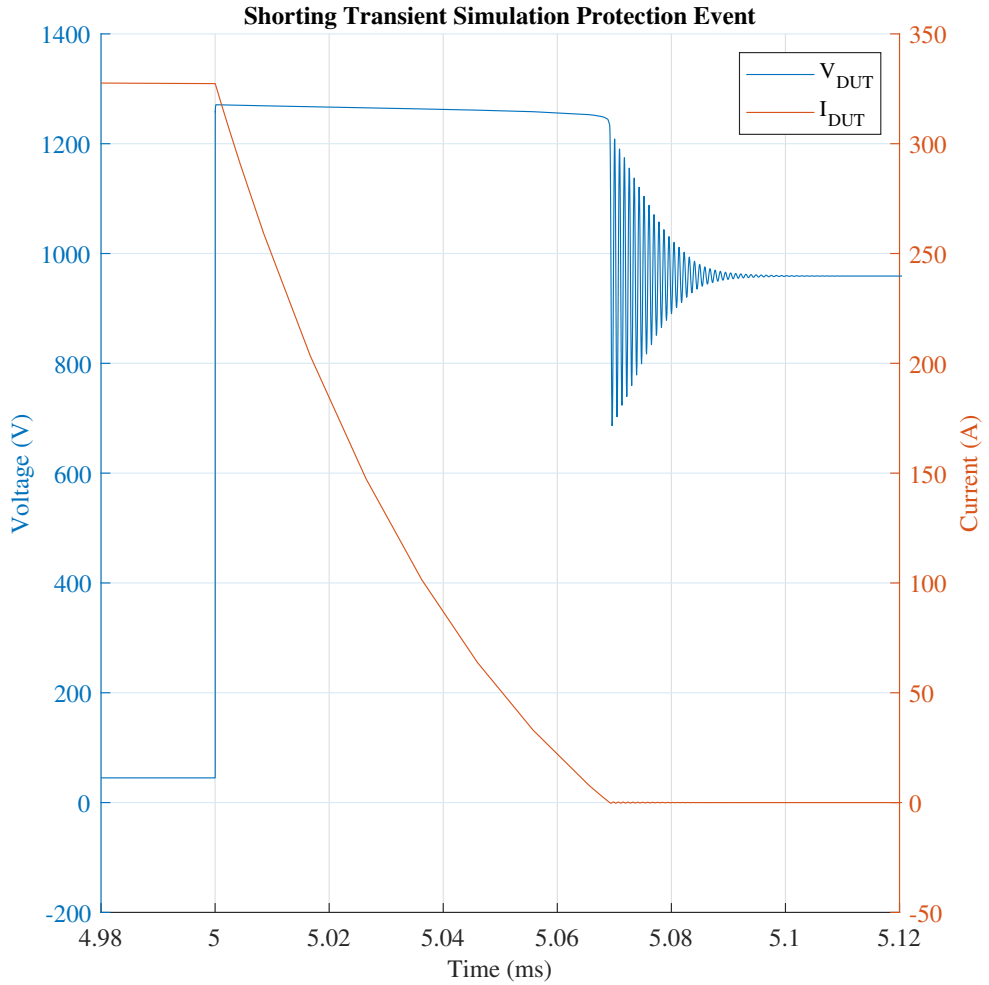


Figure 7.18: SSCB Short Circuit Test Fixture Shorting Transient Protection Event

In conclusion the simulation of the experimental test fixture was a complete success. The test fixture can be charged, discharged, or recharged in a timely manner. The steady state efficiency of the DUT can be measured using the voltage and current, as seen in Figure 7.15 at the $3ms$ to $4ms$ time point. Also, the clamping response of the EDC can be reviewed as seen at the $5ms$ time point. Overall, I have confidence in moving forward with this design of an experimental test fixture.

BIBLIOGRAPHY

[1] J. Jonnes, *Empires of Light: Edison, Tesla, Westinghouse and the Race to Electrify the World*. New York: Random House, 2003.

[2] "The westinghouse alternating current system of electrical distribution," Feb 1890.

[3] J. W. Nilsson and S. A. Riedel, *Electric Circuits*, pp. 19, 184, 187, 401. 330 Hudson Street, NY NY 10013: Pearson, 2019.

[4] F. A. T. Furfari, "Westinghouse and the ac system-1884-1895," *IEEE Industry Applications Magazine*, vol. 8, pp. 8–11, Sep. 2002.

[5] S. C. T. C. of the IEEE Power and E. Society, "Ieee standard requirements for pad-mounted, compartmental-type, self-cooled, three-phase distribution transformers, 5 mva and smaller; high voltage, 34.5 kv nominal system voltage and below; low voltage, 15 kv nominal system voltage below," *IEEE Std C57.12.34-2009 (Revision of IEEE Std C57.12.34-2004)*, pp. 1–41, 2010.

[6] K. Krause, D. Burns, S. Hutchinson, and J. Arvola, "Managing arc flash: Collaborative solutions in medium-voltage switchgear," *IEEE Industry Applications Magazine*, vol. 20, no. 1, pp. 70–78, 2014.

[7] M. Valdes, Xuhui Ren, S. Sapre, M. Trivette, and S. Meiners, "Iec ieee medium voltage distribution equipment standards, review analysis," in *2012 Petroleum and Chemical Industry Conference (PCIC)*, pp. 1–16, 2012.

[8] Dingsheng Lin and E. F. Fuchs, "Real-time monitoring of iron-core and copper losses of transformers under(non)sinusoidal operation," *IEEE Transactions on Power Delivery*, vol. 21, pp. 1333–1341, July 2006.

[9] W. W. Orr, "Aluminum and its future in power transformers," *Electrical Engineering*, vol. 78, no. 7, pp. 737–737, 1959.

[10] E. W. Tipton, "Experiences with the use of aluminum in windings for dry-type power transformers [includes discussion]," *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, vol. 74, no. 3, pp. 1201–1204, 1955.

[11] P. M. Honnell, "Note on the measurement of transformer turns-ratio," *Proceedings of the IRE*, vol. 33, pp. 808–809, Nov 1945.

[12] H. Gamo, "A general formulation of faraday's law of induction," *Proceedings of the IEEE*, vol. 67, pp. 676–677, April 1979.

[13] Dong-Keun Jeong, H. Kim, J. Baek, Ju-Yong Kim, and Hee-Je Kim, "Dual active bridge converter for energy storage system in dc microgrid," in *2016 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific)*, pp. 152–156, June 2016.

[14] D. T. Ton and M. A. Smith, "The u.s. department of energy's microgrid initiative," *The Electricity Journal*, vol. 25, pp. 84–94, October 2012.

[15] J. Roark, D. Weng, and A. Maitra, "Measuring the value of microgrids: a benefit–cost framework," *CIRED - Open Access Proceedings Journal*, vol. 2017, no. 1, pp. 1892–1894, 2017.

[16] R. M. Cuzner, K. Palaniappan, W. Sedano, N. Hoeft, and M. Qi, "Fault characterization and protective system design for a residential dc microgrid," in *2017 IEEE 6th International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 642–647, 2017.

[17] G. Boyle, *Renewable Energy Power For A Sustainable Future*, pp. 217, 394. Walton Hall, Milton Keynes MK1 6AA: Oxford, 2004.

[18] G. Krzywinski, "Integrating storage and renewable energy sources into a dc microgrid using high gain dc dc boost converters," in *2015 IEEE First International Conference on DC Microgrids (ICDCM)*, pp. 251–256, June 2015.

[19] E. Imal, M. Bagriyanik, P. Ali-Zada, K. Rajdabl, and R. Al-Zada, "A facts technique for converting existing ac transmission into hvdc (and back)," in *LESCOPE 01. 2001 Large Engineering Systems Conference on Power Engineering. Conference Proceedings. Theme: Powering Beyond 2001 (Cat. No.01ex490)*, pp. 59–62, 2001.

[20] P. Ali-Zade, O. Ustun, K. Radjabli, and K. Ali-Zade, "Two ways (ac-dc-ac or ac-ac) transmission convertion by facts technique," in *2005 IEEE Russia Power Tech*, pp. 1–4, 2005.

[21] S. Balser, S. Snakar, R. Miller, A. Rawlins, M. Israel, T. Curry, and T. Mason, "Effective grid utilization: A technical assessment and application guide," Tech. Rep. NREL/SR-5500-53696, National Renewable Energy Laboratory, 09 2012. Contract No. DE-AC36-08G028308.

[22] W. Warwick, T. Hardy, M. Hoffman, and J. Homer, "Electricity distribution system baseline report for doe quadrennial energy review," tech. rep., Pacific Northwest National Laboratory, 07 2016.

[23] S.HENRY, O.DESPOUYS, R.ADAPA, L.BARTHOLD, C.BAYFIELD, K.BELL, J.L.BINARD, A.EDRIS, P.EGROT, W.HUNG, G.IRWIN, G.KARADY, A.L'ABBATE, M.LI, M.MANCHEN, J.MESSNER, G.MIGLIAVACCA, P.MUTTIK, M.NOURIZIAN, A.PEARD, T.RAUHALA, T.SAKAI, S.SANZ, Z.SONG, S.TEMTEM, D.VAN HERTEM, and L.ZHANG, "Influence of embedded hvdc transmission on system security and ac network performance," Joint Working Group C4/B4/C1.604, cigre, France, 04 2013.

[24] J. Zhang, Z. Chen, K. Gao, and A. Yan, "A novel reactive power compensation scheme of uhv ac transmission line," in *2009 Asia-Pacific Power and Energy Engineering Conference*, pp. 1–4, March 2009.

[25] S. C. P. Transmission and Distribution, "Ieee guide for the parameter measurement of ac transmission lines," *IEEE Std 1870-2019*, pp. 1–99, July 2019.

[26] M. Farzaneh, S. Farokhi, and W. A. Chisholm, *Electrical Design of Overhead Power Transmission Lines*, ch. 4. United States of America: McGraw Hill, 213.

[27] Kyung Suk Oh, "Accurate transient simulation of transmission lines with the skin effect," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, pp. 389–396, March 2000.

[28] R. Spahr and D. Mundy, "A feasibility analysis for the transamerica generation project (tagg)-a national grid proposal," in *IEEE Power Engineering Society General Meeting, 2004.*, pp. 2277 Vol.2–, June 2004.

[29] H. M. Ayrton, "The mechanism of the electric arc," in *Proceedings of the Royal Society of London*, vol. 68, pp. 410–414, 1901.

[30] A. Mokhberdoran, A. Carvalho, H. Leite, and N. Silva, "A review on hvdc circuit breakers," in *3rd Renewable Power Generation Conference (RPG 2014)*, pp. 1–6, 2014.

[31] K. Park, H. Lee, M. Asif, and B. Lee, "Investigation of arc behavior in hvdc lc resonance circuit breaker using flexible pulsed dc-source," in *2018 Condition Monitoring and Diagnosis (CMD)*, pp. 1–5, 2018.

[32] T. A. Edison, "System of electric lighting," Oct. 1885.

[33] G. Westinghouse, "System of electric distribution," May 1886.

[34] E. Thomson, "Electric cut-out," May 1927.

[35] H. Schachtner, "Automatic electric circuit-breaker," Nov. 1893.

[36] M. Abedrabbo, W. Leterme, and D. Van Hertem, "Analysis and enhanced topologies of active-resonance dc circuit breaker," in *2017 19th European Conference on Power Electronics and Applications (EPE'17 ECCE Europe)*, pp. P.1–P.10, 2017.

[37] X. Pei, O. Cwikowski, D. S. Vilchis-Rodriguez, M. Barnes, A. C. Smith, and R. Shuttleworth, "A review of technologies for mvdc circuit breakers," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pp. 3799–3805, 2016.

[38] T.Eriksson, M.Backman, and S.Halen, "A low loss mechanical hvdc breaker for hvdc grid applications," in *CIGRE SESSION PAPERS & PROCEEDINGS*, no. 303 in B4, 2014.

[39] Y. Bingjian, G. Yang, W. Xiaoguang, H. Zhiyuan, C. Longlong, and S. Yunhai, "A hybrid circuit breaker for dc-application," in *2015 IEEE First International Conference on DC Microgrids (ICDCM)*, pp. 187–192, 2015.

[40] J. S. Morton, "Coil-less cold-cathode arc chutes for high-speed d.c. circuit breakers for use on traction systems," *IEE Proceedings B - Electric Power Applications*, vol. 127, no. 1, pp. 34–45, 1980.

[41] B. Pauli, G. Mauthe, E. Ruoss, G. Ecklin, J. Porter, and J. Vithayathil, "Development of a high current hvdc circuit breaker with fast fault clearing capability," *IEEE Transactions on Power Delivery*, vol. 3, no. 4, pp. 2072–2080, 1988.

[42] M. Mobarrez, M. G. Kashani, S. Bhattacharya, and R. Adapa, "Comparative study of dc circuit breakers using realtime simulations," in *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3736–3742, 2014.

[43] W. Wen, Y. Huang, Y. Sun, J. Wu, M. Al-Dweikat, and W. Liu, "Research on current commutation measures for hybrid dc circuit breakers," *IEEE Transactions on Power Delivery*, vol. 31, no. 4, pp. 1456–1463, 2016.

[44] X. Song, C. Peng, and A. Q. Huang, "A medium-voltage hybrid dc circuit breaker, part i: Solid-state main breaker based on 15 kv sic emitter turn-off thyristor," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 5, pp. 278–288, March 2017.

[45] A. Shukla and G. D. Demetriades, "A survey on hybrid circuit-breaker topologies," *IEEE Transactions on Power Delivery*, vol. 30, no. 2, pp. 627–641, 2015.

[46] R. Rodrigues, Y. Du, A. Antoniazzi, and P. Cairoli, "A review of solid-state circuit breakers," *IEEE Transactions on Power Electronics*, vol. 36, no. 1, pp. 364–377, 2021.

[47] C. Peng, X. Song, A. Q. Huang, and I. Husain, "A medium-voltage hybrid dc circuit breaker—part ii: Ultrafast mechanical switch," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 5, no. 1, pp. 289–296, 2017.

[48] Z. J. Shen, Z. Miao, and A. M. Roshandeh, "Solid state circuit breakers for dc micrgrids: Current status and future trends," in *2015 IEEE First International Conference on DC Microgrids (ICDCM)*, pp. 228–233, 2015.

[49] J. Langston, K. Schoder, M. Sloderbeck, M. Steurer, and A. Rockhill, "Testing operation and coordination of dc solid state circuit breakers," in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3445–3452, 2018.

[50] R. Su, P. Y. Chiang, J. Gong, T. Huang, C. L. Tsai, C. C. Chou, and C. M. Liu, "Experimental results of on-state resistance reduction by sti fingers in ldmosfet," *IEEE Electron Device Letters*, vol. 30, no. 2, pp. 192–194, 2009.

[51] M. Kempkes, I. Roth, and M. Gaudreau, "Solid-state circuit breakers for medium voltage dc power," in *2011 IEEE Electric Ship Technologies Symposium*, pp. 254–257, 2011.

[52] L. Qi, P. Cairoli, Z. Pan, C. Tschida, Z. Wang, V. R. Ramanan, L. Raciti, and A. Antoniazzi, "Solid-state circuit breaker protection for dc shipboard power systems: Breaker design, protection scheme, validation testing," *IEEE Transactions on Industry Applications*, vol. 56, no. 2, pp. 952–960, 2020.

[53] D. He, Z. Shuai, Z. Lei, W. Wang, X. Yang, and Z. J. Shen, "A sic jfet-based solid state circuit breaker with digitally controlled current-time profiles," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 7, no. 3, pp. 1556–1565, 2019.

[54] X. Liao, H. Li, R. Yao, Z. Huang, and K. Wang, "Voltage overshoot suppression for sic mosfet-based dc solid-state circuit breaker," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 9, no. 4, pp. 649–660, 2019.

[55] D. Jovcic, M. Hajian, and L. Zhang, "Dc transmission grids with fault tolerant lcl vsc converters and mechanical dc circuit breakers," in *10th IET International Conference on AC and DC Power Transmission (ACDC 2012)*, pp. 1–6, 2012.

[56] T. Augustin, M. Becerra, and H. P. Nee, "Enhanced active resonant dc circuit breakers based on discharge closing switches," *IEEE Transactions on Power Delivery*, pp. 1–1, 2020.

[57] L. Ängquist, S. Norrga, and T. Modéer, "A new dc breaker with reduced need for semiconductors," in *2016 18th European Conference on Power Electronics and Applications (EPE'16 ECCE Europe)*, pp. 1–9, 2016.

[58] Y. Wang and R. Marquardt, "Future hvdc-grids employing modular multilevel converters and hybrid dc-breakers," in *2013 15th European Conference on Power Electronics and Applications (EPE)*, pp. 1–8, 2013.

[59] C. C. Davidson, R. S. Whitehouse, C. D. Barker, J. . Dupraz, and W. Grieshaber, "A new ultra-fast hvdc circuit breaker for meshed dc networks," in *11th IET International Conference on AC and DC Power Transmission*, pp. 1–7, 2015.

[60] A. K. Kale and M. P. Thakre, "Integration of proactive hybrid circuit breaker with current flow controller for bipolar hvdc grid," in *2019 International Conference on Power Electronics Applications and Technology in Present Energy Scenario (PETPES)*, pp. 1–6, 2019.

[61] F. Xu, H. Yu, Y. Lu, P. Qiu, K. Tong, J. Xuan, Q. Chen, X. Huang, and D. Jiang, "Topology, control and fault analysis of a new type hvdc breaker for hvdc systems," in *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pp. 1959–1964, 2016.

[62] J. Zhu, x. guo, j. yin, Q. Huo, and T. Wei, "Basic topology, modeling and evaluation of a novel hybrid dc breaker for hvdc grid," *IEEE Transactions on Power Delivery*, pp. 1–1, 2020.

[63] W. Pusorn, W. Srisongkram, K. Chiangchin, and K. Bhumkittipich, "Solid state circuit breaker using insulated gate bipolar transistor for distribution system protection," in *2014 International Electrical Engineering Congress (iEECON)*, pp. 1–4, 2014.

[64] Q. Lu, Y. Zhanqing, H. Songbo, L. Jiapeng, Z. Wenpeng, C. Zhengyu, Z. Jie, X. Ning, and Z. Rong, "Design and analysis of a 375v/5ka solid state dc circuit breaker based on igct," in *2018 IEEE International Power Electronics and Application Conference and Exposition (PEAC)*, pp. 1–5, 2018.

[65] K. Sano and M. Takasaki, "A surgeless solid-state dc circuit breaker for voltage-source-converter-based hvdc systems," *IEEE Transactions on Industry Applications*, vol. 50, no. 4, pp. 2690–2699, 2014.

[66] S. Li, C. Zhao, and J. Xu, "A new topology for current-limiting solid-state hvdc circuit breaker," in *2016 IEEE 2nd Annual Southern Power Electronics Conference (SPEC)*, pp. 1–5, 2016.

[67] K. A. Corzine and R. W. Ashton, "A new z-source dc circuit breaker," *IEEE Transactions on Power Electronics*, vol. 27, no. 6, pp. 2796–2804, 2012.

[68] K. A. Corzine and R. W. Ashton, "Structure and analysis of the z-source mvdc breaker," in *2011 IEEE Electric Ship Technologies Symposium*, pp. 334–338, 2011.

[69] W. Song, N. An, and Y. Wang, "A novel bidirectional t-source dc circuit breaker for dc microgrids," in *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1540–1545, 2019.

[70] H. Al-khafaf and J. Asumadu, "Y-source bi-directional dc circuit breaker," in *2018 International Power Electronics Conference (IPEC-Niigata 2018 -ECCE Asia)*, pp. 3445–3449, 2018.

[71] J. Hayes, K. George, P. Killeen, B. McPherson, K. J. Olejniczak, and T. R. McNutt, "Bidirectional, sic module-based solid-state circuit breakers for 270 vdc mea/aea systems," in *2016 IEEE 4th Workshop on Wide Bandgap Power Devices and Applications (WiPDA)*, pp. 70–77, 2016.

[72] United SiC, *SiC FET*, December 2018. Rev. B.

[73] CREE, *Silicon Carbide N-Channel Enhancement Mode Power MOSFET*, February 2019. Rev. D.

[74] United SiC, *SiC FET*, December 2019. Rev. D.

[75] Infineon Technologies, *Trenchstop IGBT 6*, May 2018. Rev. 2.1.

[76] R. W. Erickson and D. Maksimović, *Fundamentals of Power Electronics*, pp. 11,66,68,72. 233 Spring Street, NY NY 10013: Springer, 1999.

[77] S. Hazra, A. De, L. Cheng, J. Palmour, M. Schupbach, B. A. Hull, S. Allen, and S. Bhattacharya, "High switching performance of 1700-v, 50-a sic power mosfet over si igbt/bimosfet for advanced power conversion applications," *IEEE Transactions on Power Electronics*, vol. 31, no. 7, pp. 4742–4754, 2016.

[78] S. D. Umans, *Fitzgerald & Kingsley's Electric Machinery*, pp. 304,677. 1221 Avenue of the Americas, NY NY 10020: McGraw Hill, 2014.

[79] G. J. Van Wylen and R. E. Sonntag, *Fundamentals of Classical Thermodynamics*, p. 90. New York, London, Sydney: John Wiley & Sons Inc., 1968.

[80] Y. A. Cengal and A. J. Ghajar, *Heat and Mass Transfer Fundamentals & Applications*, pp. 10, 18, 29. 2 Penn Plaza, New York, NY 10121: McGraw Hill Education, 2015.

[81] M. J. Moran, H. N. Shapiro, D. D. Boettner, and M. B. Bailey, *Fundamentals of Engineering Thermodynamics*, p. 56. 111 River Street, Hoboken, NJ 07030-577: John Wiley & Sons Inc., 2011.

[82] N. Mohan, *First Course on Power Electronics and Drives*. Minneapolis, MN 55414: MNPERE, 2003.

[83] S. Lee, "Optimum design and selection of heat sinks," *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part A*, vol. 18, no. 4, pp. 812–817, 1995.

[84] Seri Lee, "Optimum design and selection of heat sinks," in *Proceedings of 1995 IEEE/CPMT 11th Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM)*, pp. 48–54, 1995.

[85] United SiC, *SiC FET*, December 2019. Rev. B.

[86] Infineon Technologies, *CoolSiC SiC Trench MOSFET*, December 2019. Rev. 2.1.

[87] Littelfuse Inc., *Enhancement-mode SiC MOSFET*, October 2018. Rev. 0.2.

[88] Microsemi, *Silicon Carbide N-Channel Power MOSFET*, February 2020. Rev. A.

[89] Rohm Semiconductor, *Silicon Carbide N-Channel Power MOSFET*, June 2018. Rev. 005.

[90] CREE, *Silicon Carbide N-Channel Enhancement Mode Power MOSFET*, February 2019. Rev. A.

[91] Infineon Technologies, *Trenchstop IGBT 2*, December 2013. Rev. 2.2.

[92] G. Roberts and A. Sedra, *Spice*. New York: Oxford University Press, 1997.

[93] Ohmite, *SOT-227, Thermal Interface Material for Heatsinkable devices.* rev 7/20-3.

[94] Wakefield-vette, *TO-247, ulTIMiFlux Dielectric Phase Change Thermal Material.*

[95] Littlefuse, *Transient Voltage Suppression Diodes*, October 2019. Axial Leaded – 10kA >AK10-Y series.

[96] Littlefuse, *Transient Voltage Suppression Diodes*, November 2015. Axial Leaded – 3kA >AK3 series.

[97] Littlefuse, *Transient Voltage Suppression Diodes*, November 2015. Axial Leaded – 6kA >AK6 series.

[98] Bourns, *PTVS10-xxxC-TH Series High Voltage, High Current TVS Diodes*, November 2015. C1753 05/17/18R.

[99] Bourns, *PTVS3-xxxC-TH Series High Voltage, High Current TVS Diodes*, April 2017. C1753 05/17/18R.

[100] Jian-Bo Yang and M. G. Singh, "An evidential reasoning approach for multiple-attribute decision making with uncertainty," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 1, pp. 1–18, 1994.

APPENDIX A
ACRONYMS AND TERMINOLOGY

A-IGCT asymmetrical integrated gate commutated thyristor 27
AC alternating current 1
DC direct current 1
DOE U.S. Department of Energy 5
DUT device under test 142
EDC energy dissipation circuit 14
ERCOT Electric Reliability Council of Texas 7
EV electric vehicle 5
FET field-effect transistor 38
GDT gas discharge tube 58
HV high-voltage 2
HVDC high voltage direct current 6
IEC International Electrotechnical Commission 2
IEEE Institute of Electrical and Electronic Engineers 2
IEEE Institute of Electrical and Electronics Engineers 2
IGBT insulated gate bipolar transistor 27
JFET junction field-effect transistor 30
LED light-emitting diode 5
LV low-voltage 2
MCB mechanical circuit breaker 21
MOSFET metal-oxide semiconductor field-effect transistor 37
MOV metal-oxide varistor 28
MV medium-voltage 2
NEMA National Electrical Manufacturers Association 2
NERC North American Electric Reliability Corporation 7
NFPA National Fire Protection Association 2
NREL National Renewable Energy Laboratory 7
PCB printed circuit board 47
PV photo-voltaic 6
RB-IGCT reverse blocking integrated gate commutated thyristor 27
RCB resonant circuit breaker 23
Si silicon 39
SiC silicon-carbide 30
SPICE Simulation Program with Integrated Circuit Emphasis 45

APPENDIX B
SCHEMATICS

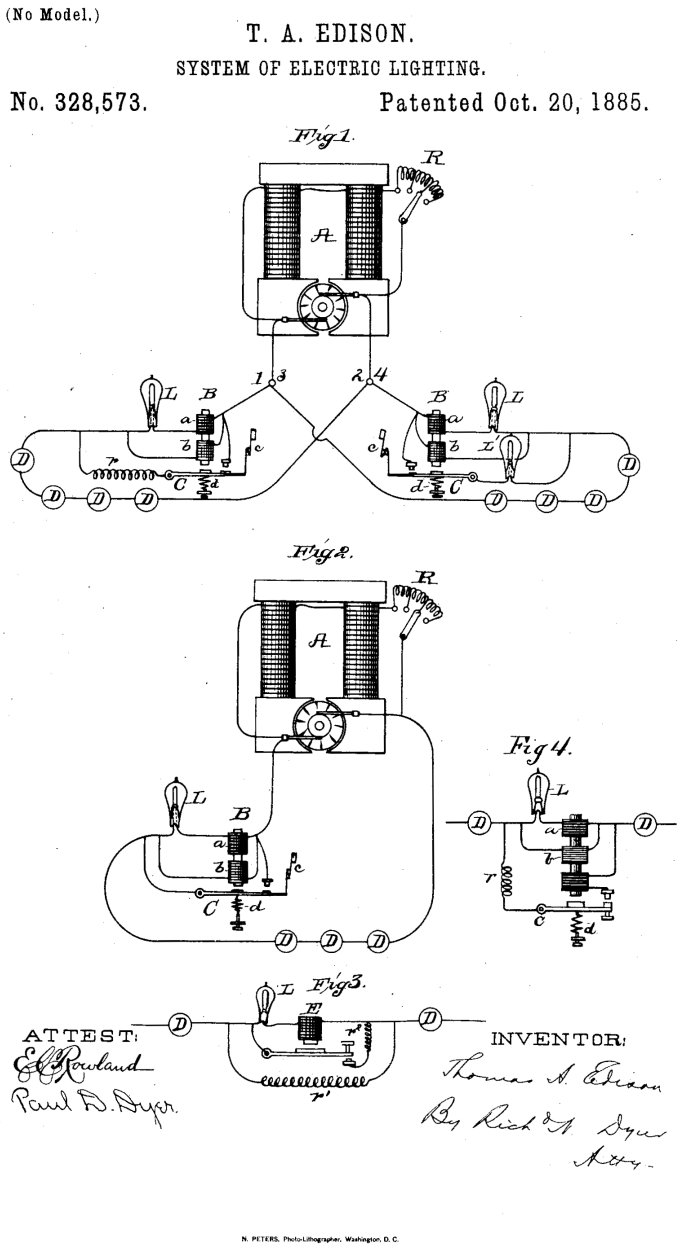
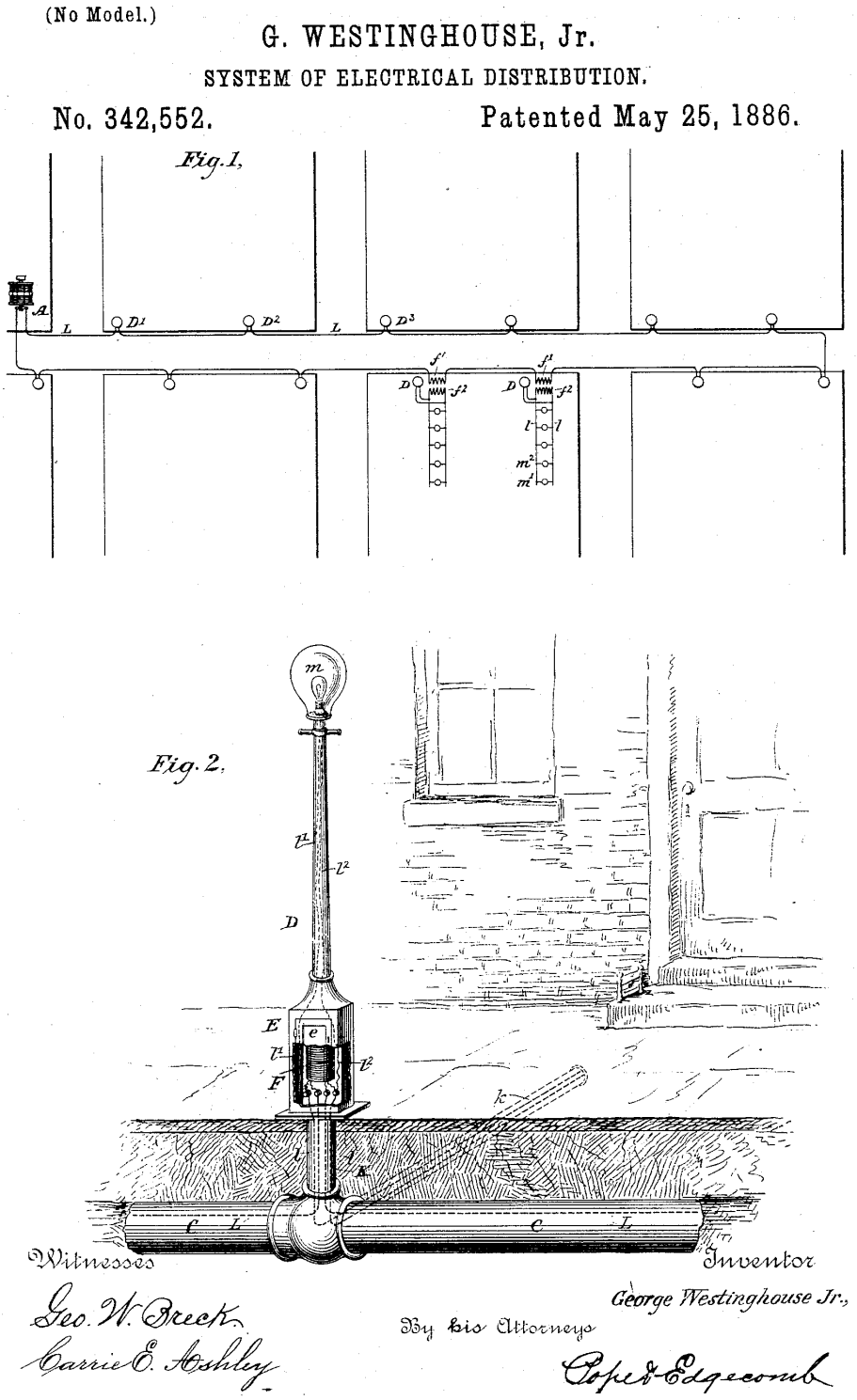


Figure B.1: System Of Electric Lighting Patent Figure [32]


(No Model.)

G. WESTINGHOUSE, Jr.

SYSTEM OF ELECTRICAL DISTRIBUTION.

No. 342,552. Patented May 25, 1886.

Fig. 1.

Fig. 2.

Witnesses

Geo. W. Breck.

Carrie E. Ashley.

Inventor

George Westinghouse Jr.,

By his Attorneys

Pope & Edgecomb

Figure B.2: System of Electrical Distribution Patent Figure [33]

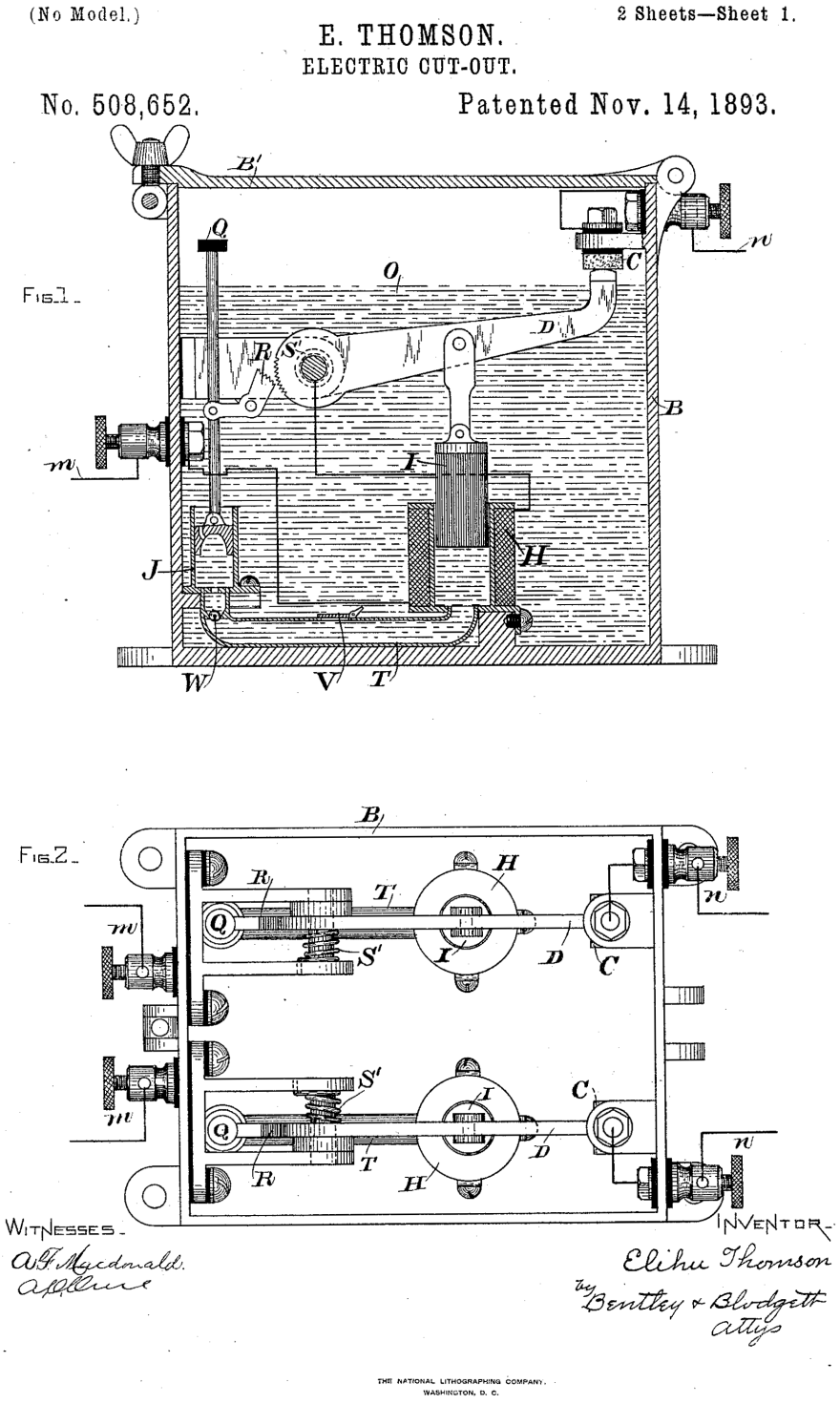


Figure B.3: Early Automatic Electrical Cut-Out Patent Figure 1 of 2 [34]

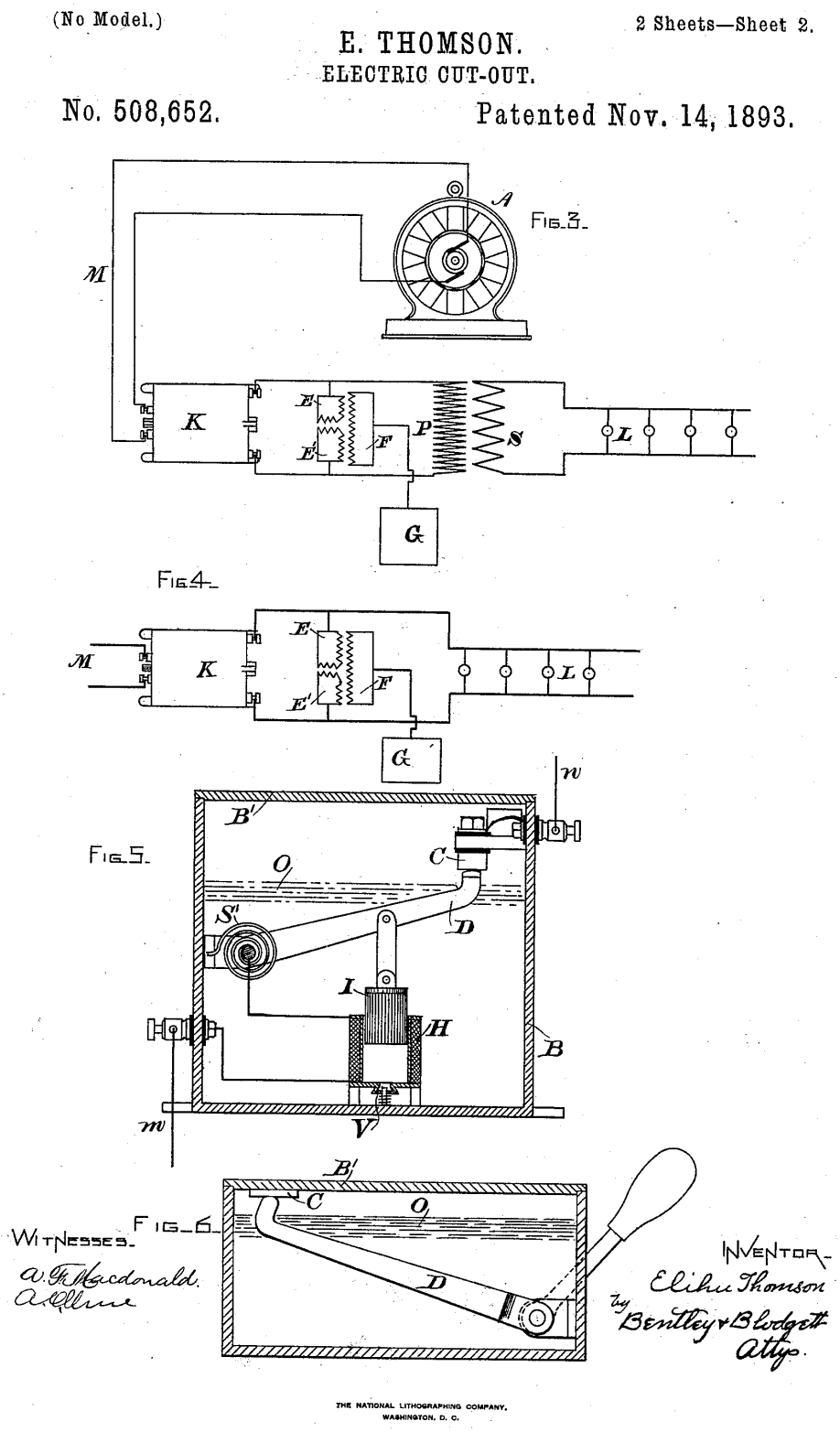


Figure B.4: Early Automatic Electrical Cut-Out Patent Figure 2 of 2 [34]

May 24, 1927. 1,629,640

H. SCHACHTNER

AUTOMATIC ELECTRIC CIRCUIT BREAKER

Filed Aug. 19, 1926 3 Sheets-Sheet 1

FIG. 1

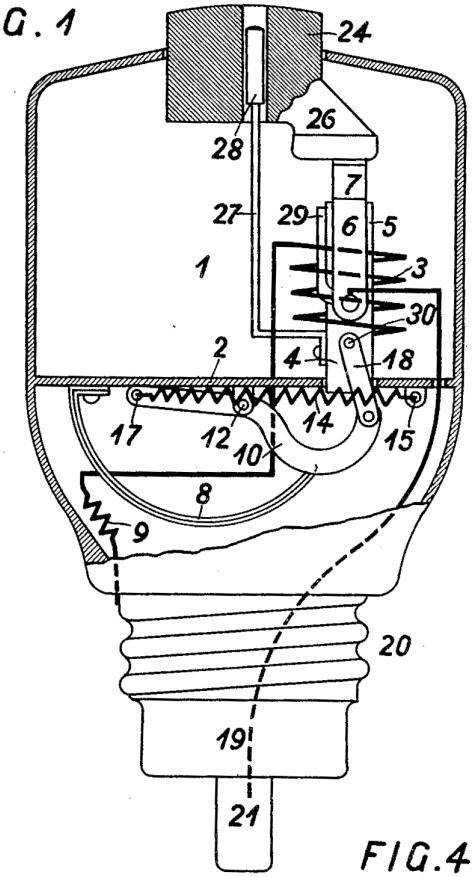
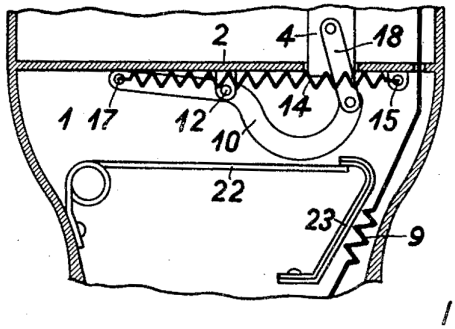


FIG. 4



INVENTOR

H. Schachtner

Figure B.5: Automatic Electric Circuit-Breaker Patent Figure 1 of 3 [35]

May 24, 1927. 1,629,640

H. SCHACHTNER

AUTOMATIC ELECTRIC CIRCUIT BREAKER

Filed Aug. 19, 1926 3 Sheets-Sheet 2

FIG. 2

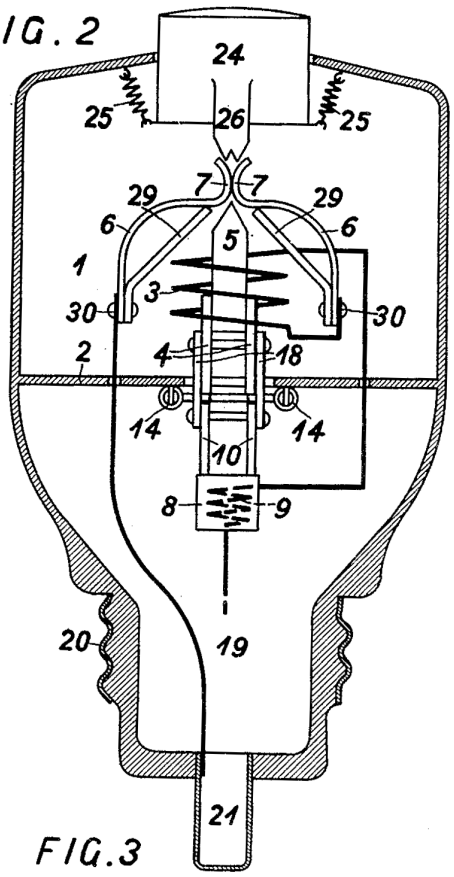
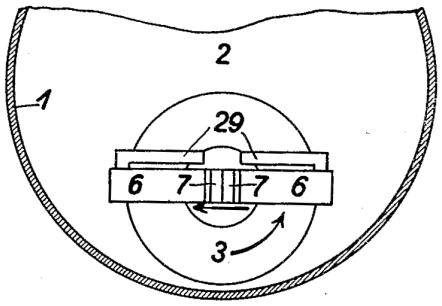


FIG. 3



INVENTOR

Figure B.6: Automatic Electric Circuit-Breaker Patent Figure 2 of 3 [35]

May 24, 1927. 1,629,640

H. SCHACHTNER

AUTOMATIC ELECTRIC CIRCUIT BREAKER

Filed Aug. 19, 1926 3 Sheets-Sheet 3

FIG. 5

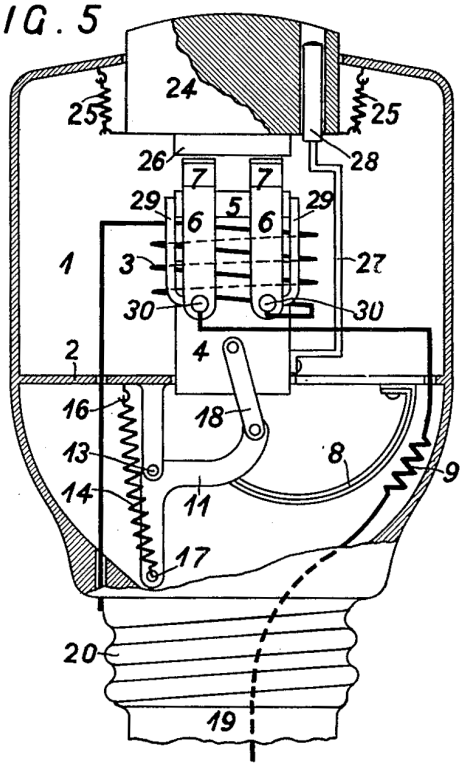
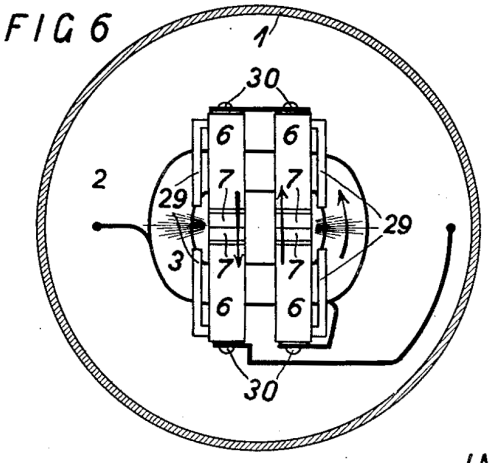


FIG 6



INVENTOR

Figure B.7: Automatic Electric Circuit-Breaker Patent Figure 3 of 3 [35]

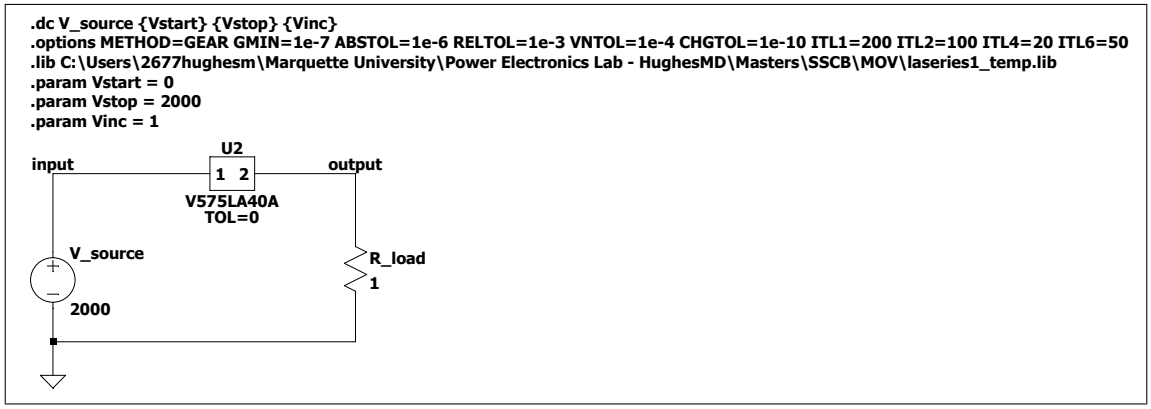


Figure B.8: MOV Littelfuse V575LA40A I-V Curve Circuit

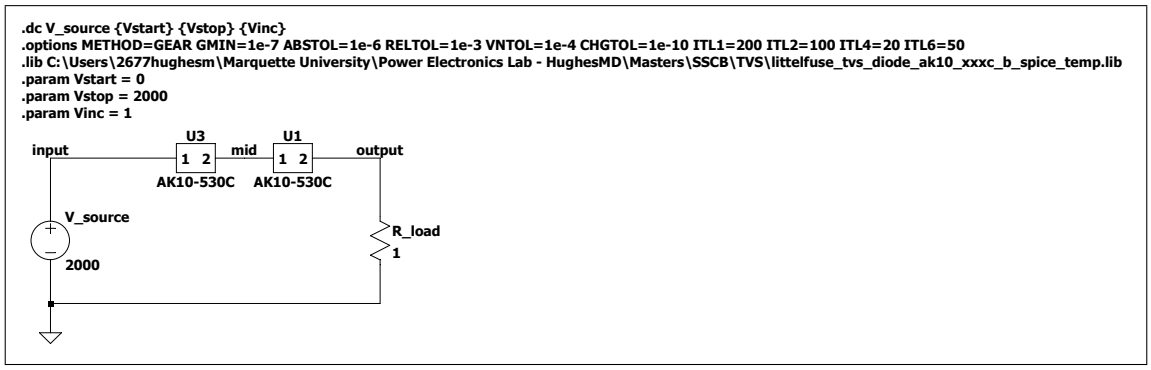


Figure B.9: TVS Diode Littelfuse AK10530C I-V Curve Circuit

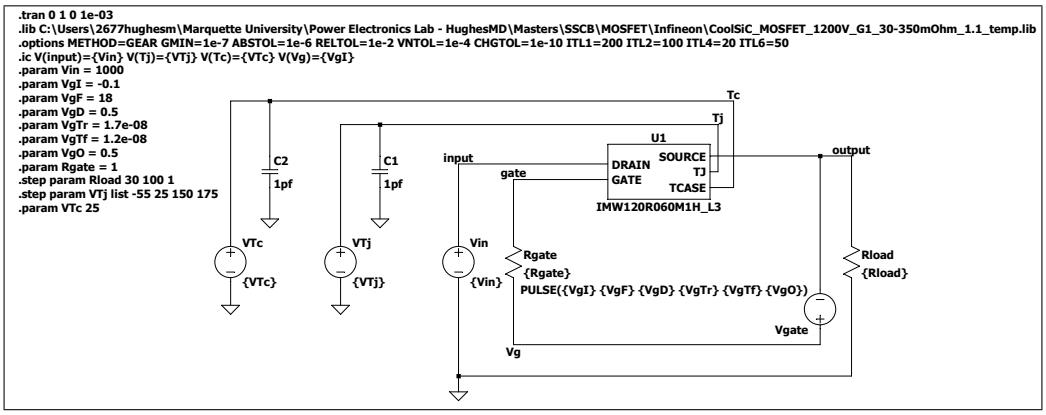


Figure B.10: Unidirectional Efficiency of Device 1

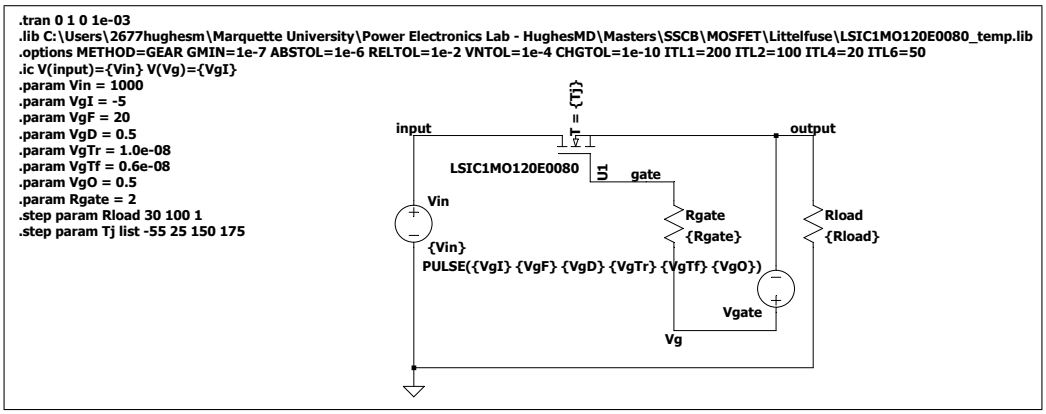


Figure B.11: Unidirectional Efficiency of Device 2

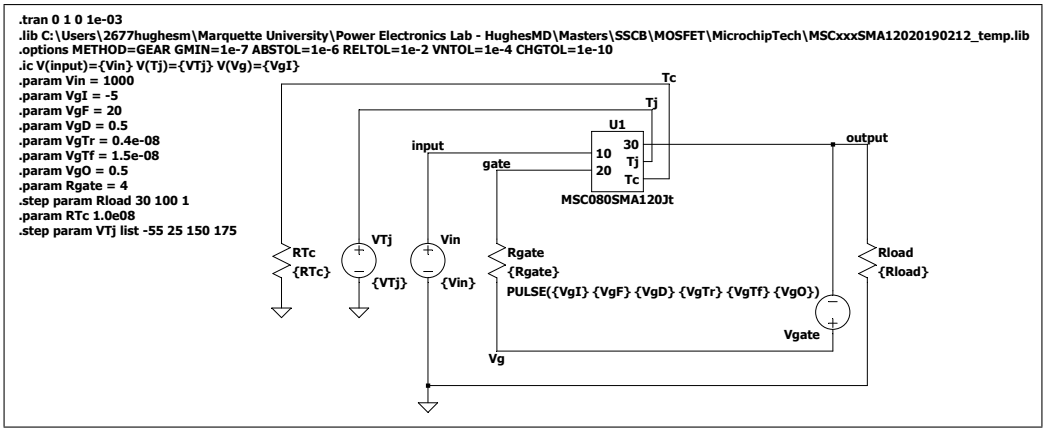


Figure B.12: Unidirectional Efficiency of Device 3

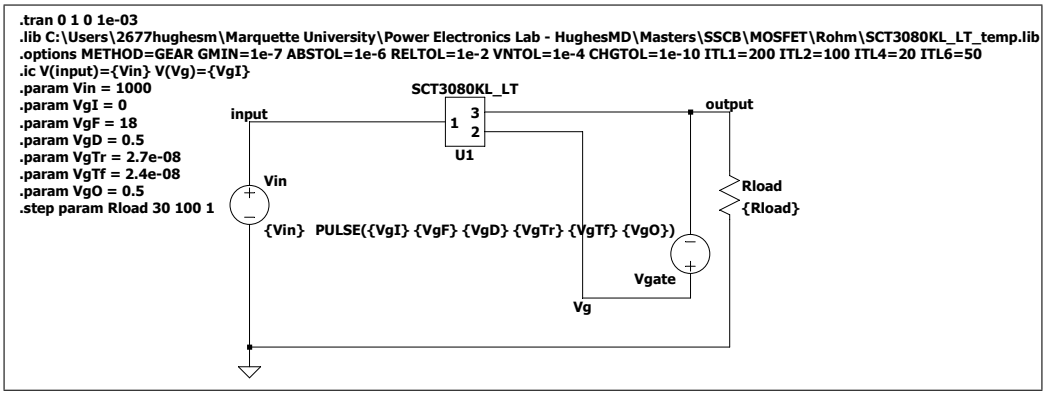


Figure B.13: Unidirectional Efficiency of Device 4

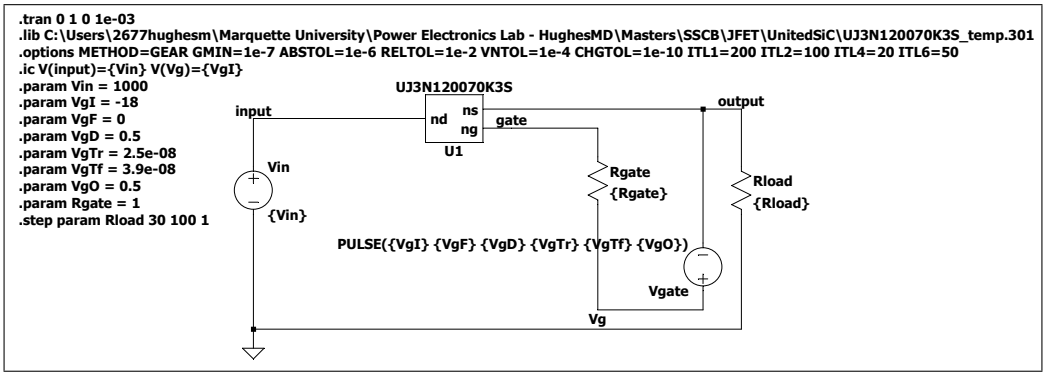


Figure B.14: Unidirectional Efficiency of Device 5

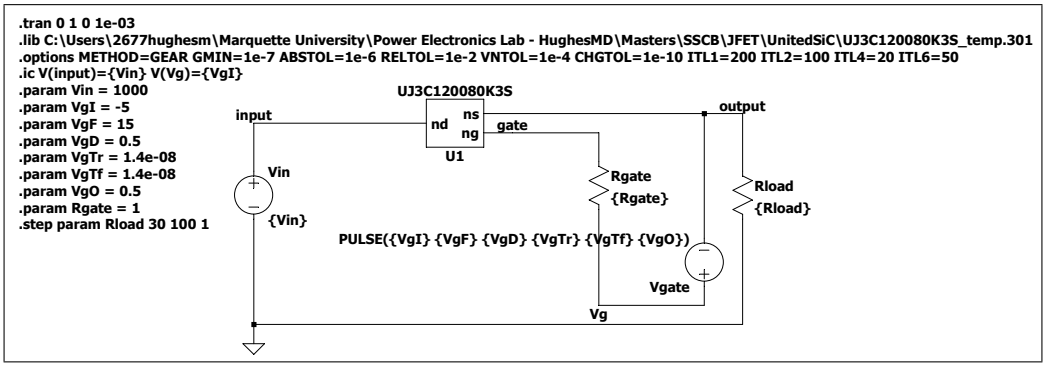


Figure B.15: Unidirectional Efficiency of Device 6

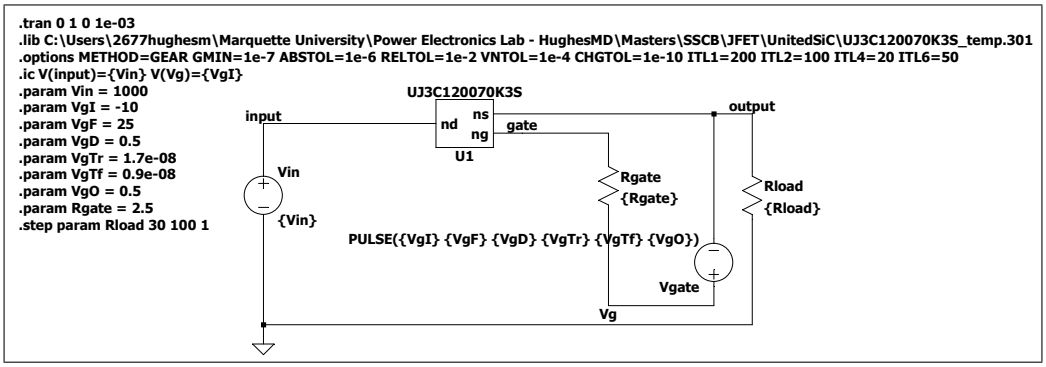


Figure B.16: Unidirectional Efficiency of Device 7

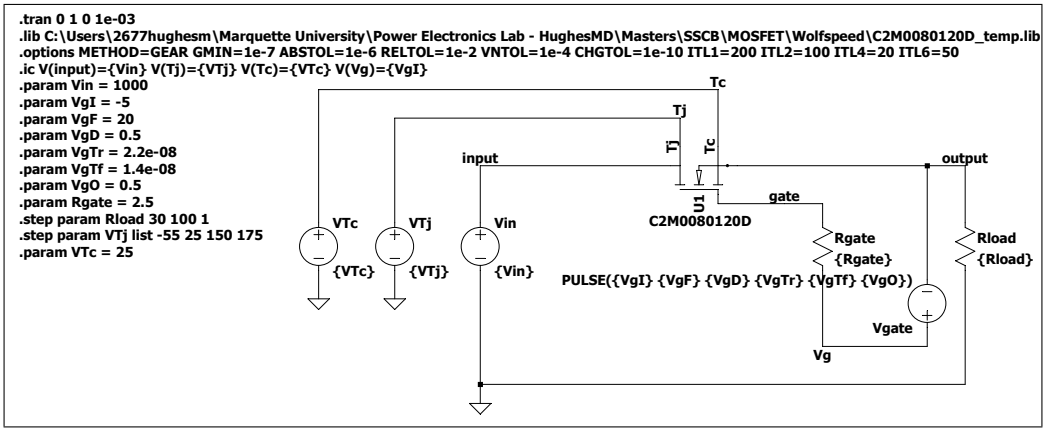


Figure B.17: Unidirectional Efficiency of Device 8

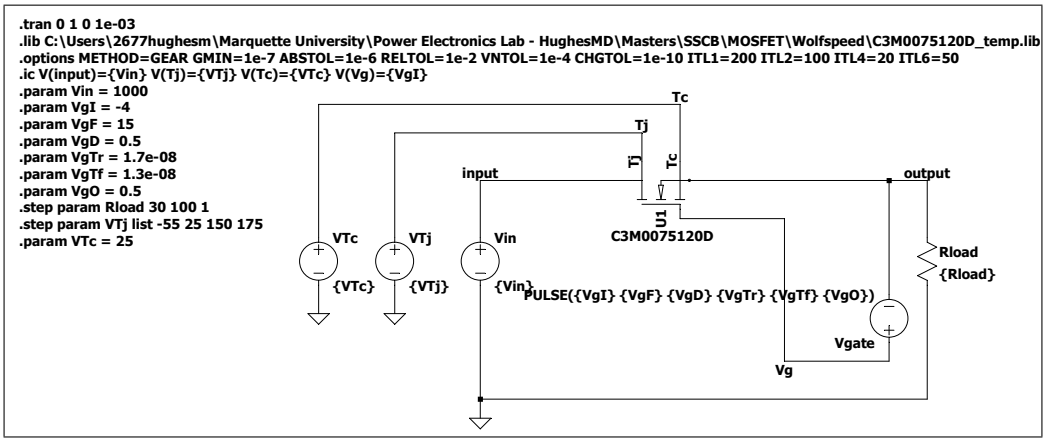


Figure B.18: Unidirectional Efficiency of Device 9

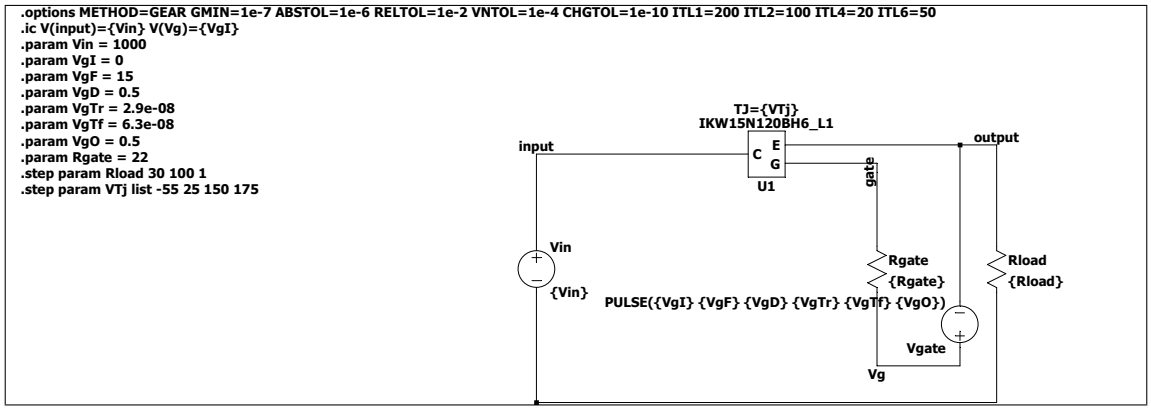


Figure B.19: Unidirectional Efficiency of Device 10

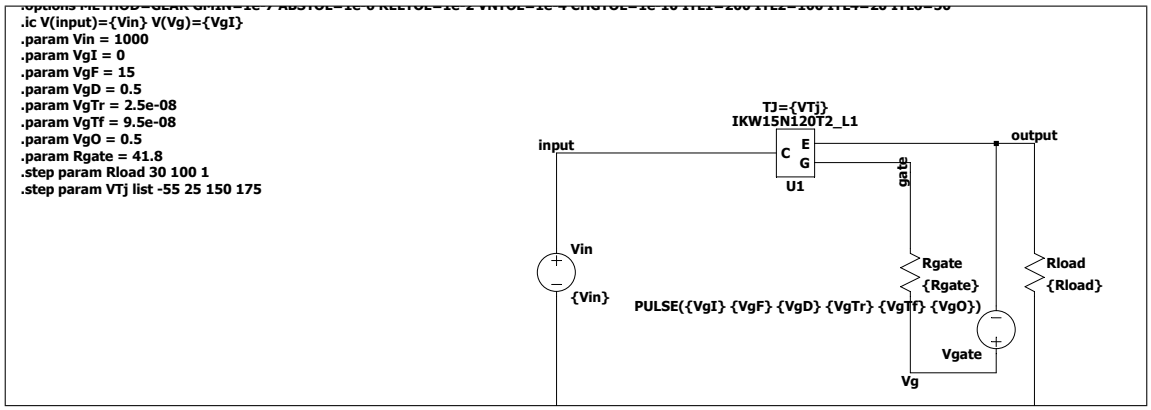


Figure B.20: Unidirectional Efficiency of Device 11

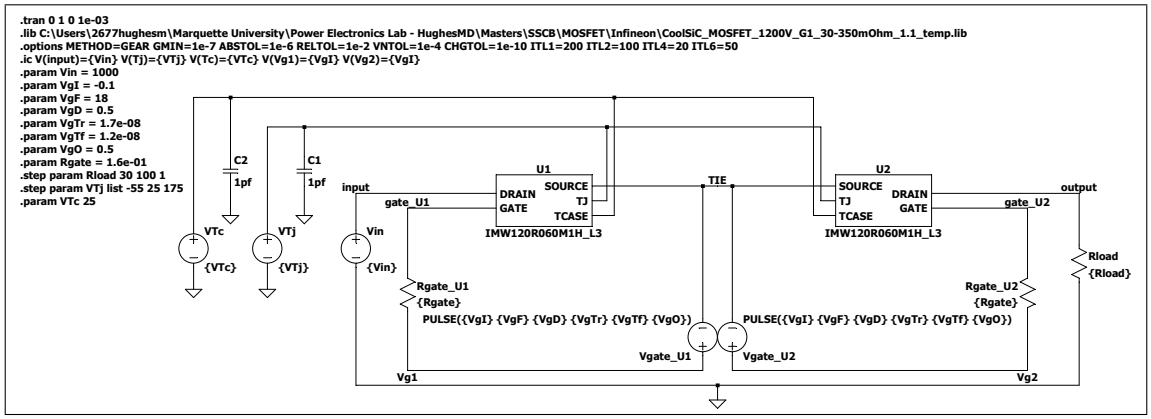


Figure B.21: Bidirectional Efficiency of Device 1

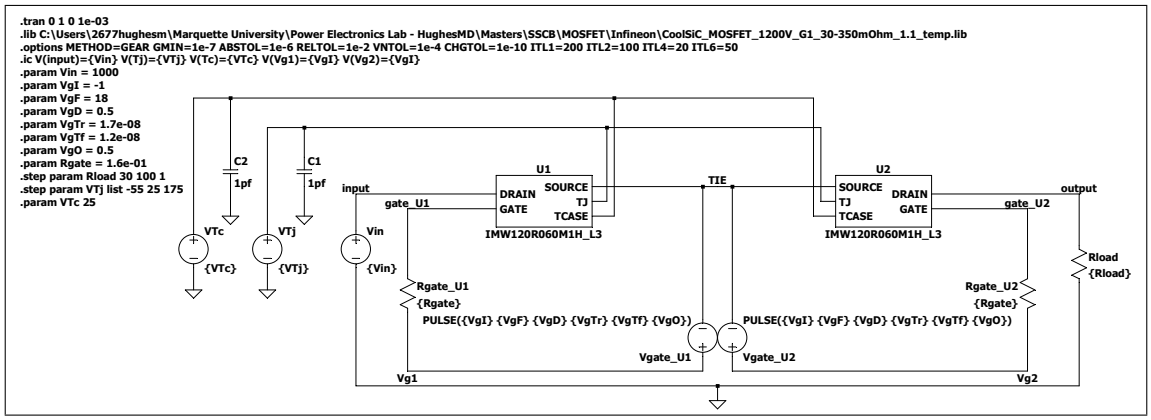


Figure B.22: Bidirectional Efficiency of Device 2

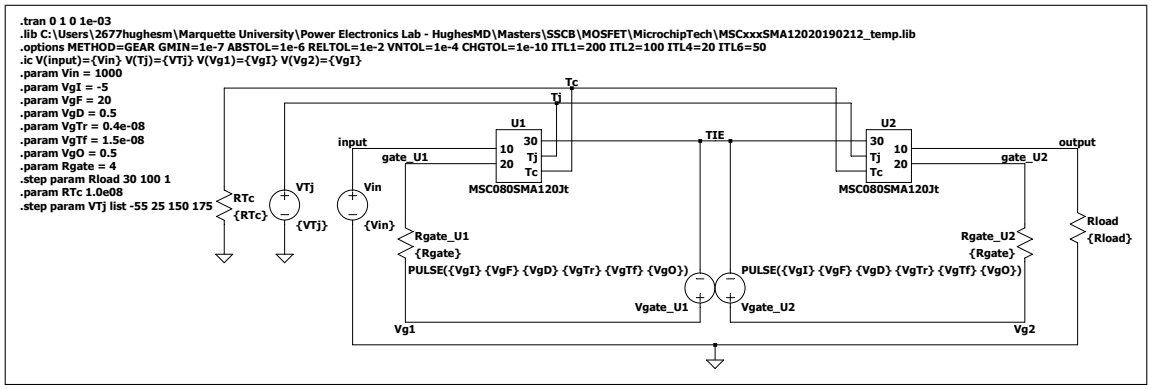


Figure B.23: Bidirectional Efficiency of Device 3

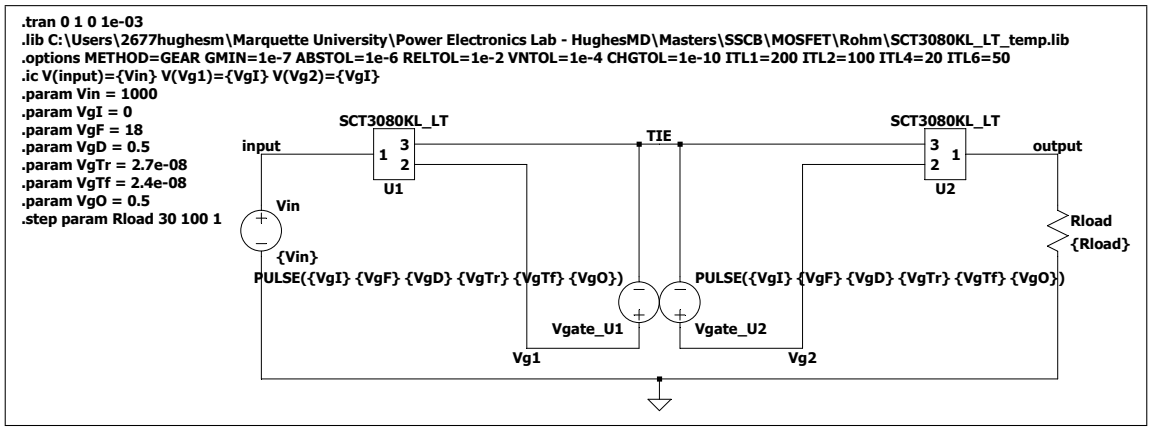


Figure B.24: Bidirectional Efficiency of Device 4

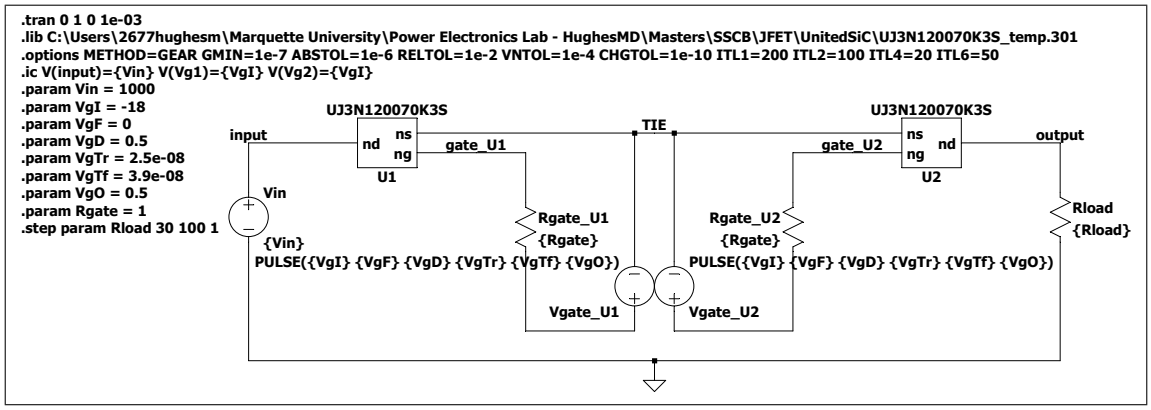


Figure B.25: Bidirectional Efficiency of Device 5

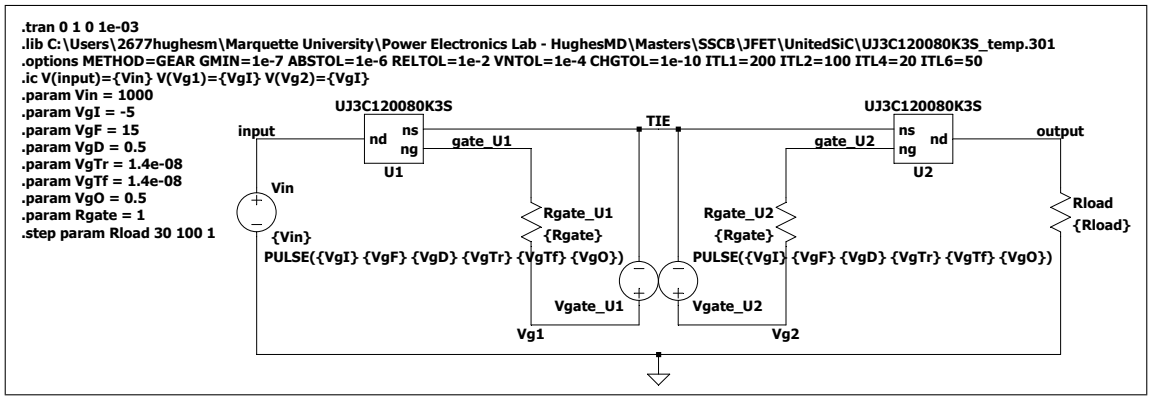


Figure B.26: Bidirectional Efficiency of Device 6

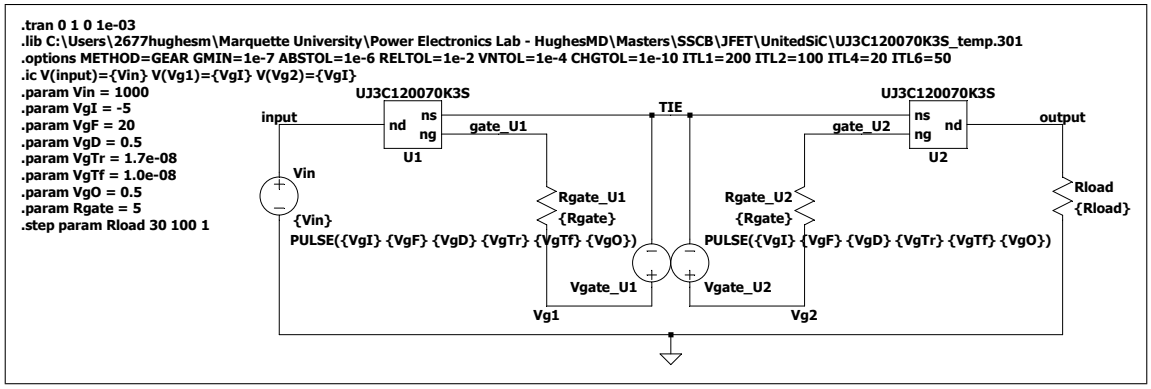


Figure B.27: Bidirectional Efficiency of Device 7

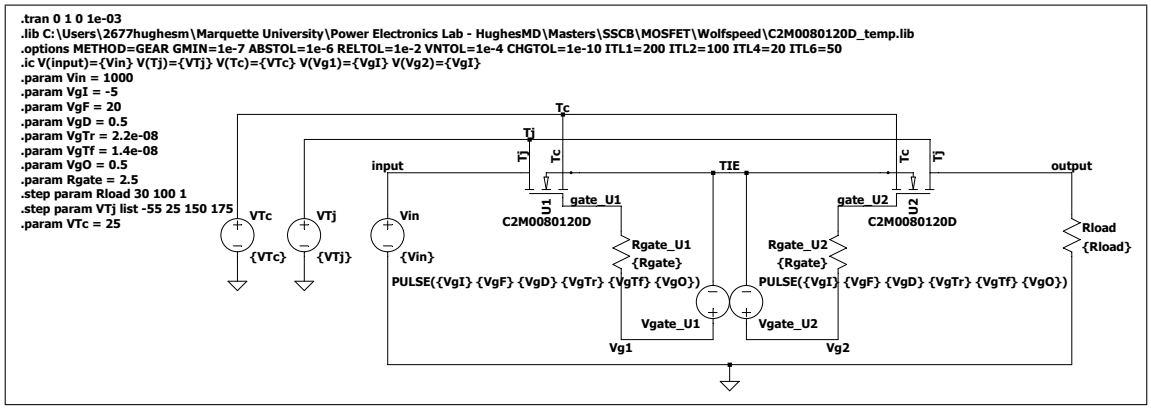


Figure B.28: Bidirectional Efficiency of Device 8

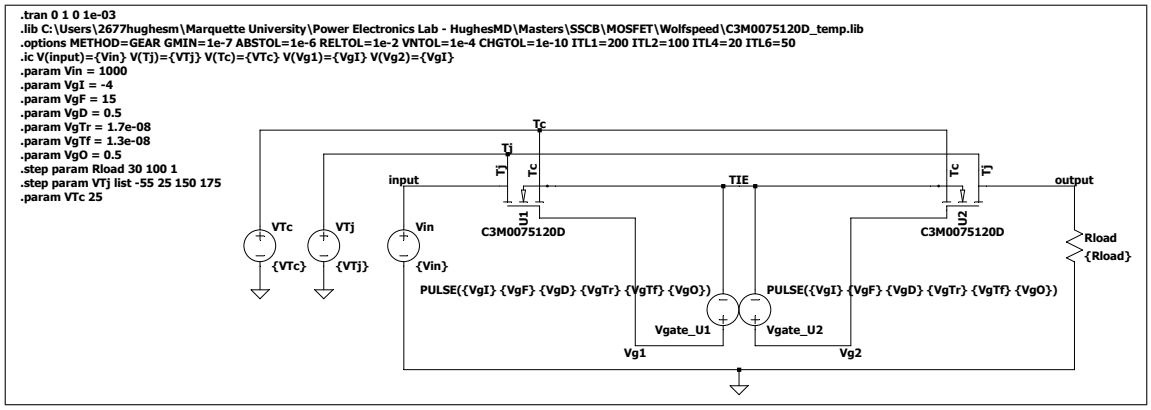


Figure B.29: Bidirectional Efficiency of Device 9

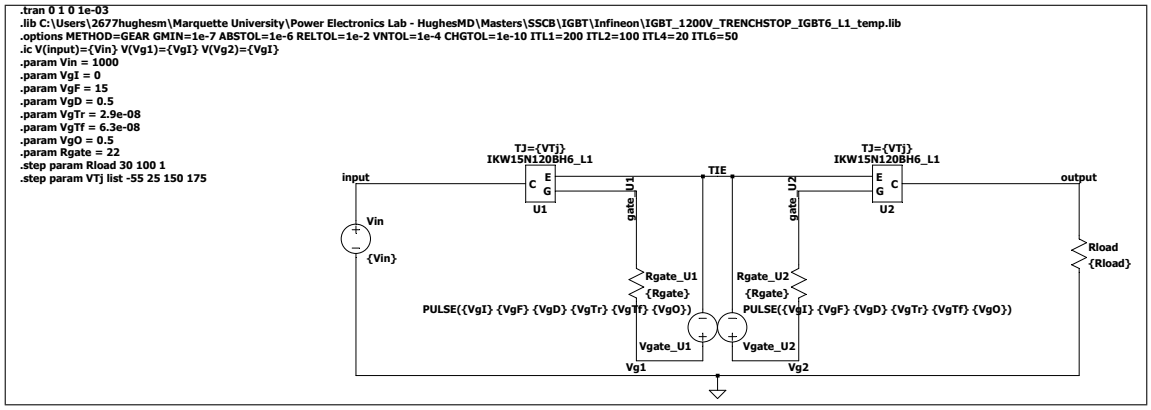


Figure B.30: Bidirectional Efficiency of Device 10

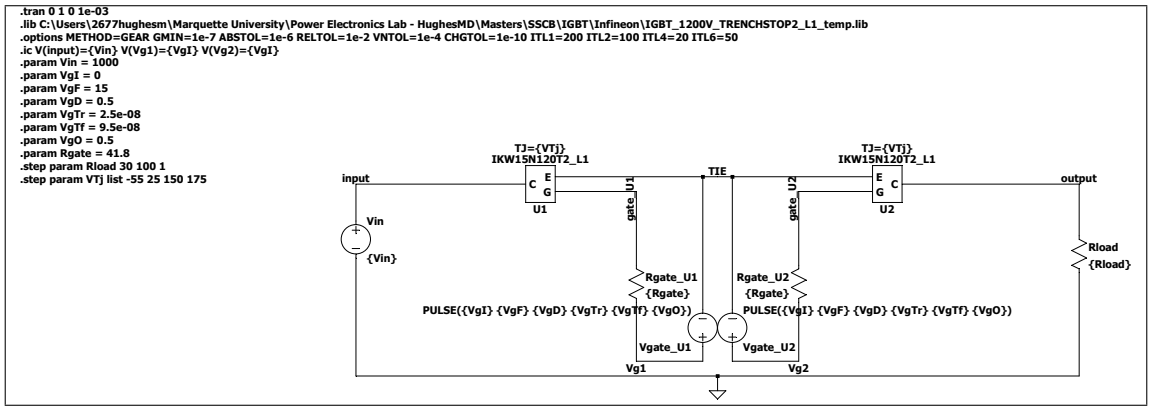


Figure B.31: Bidirectional Efficiency of Device 11

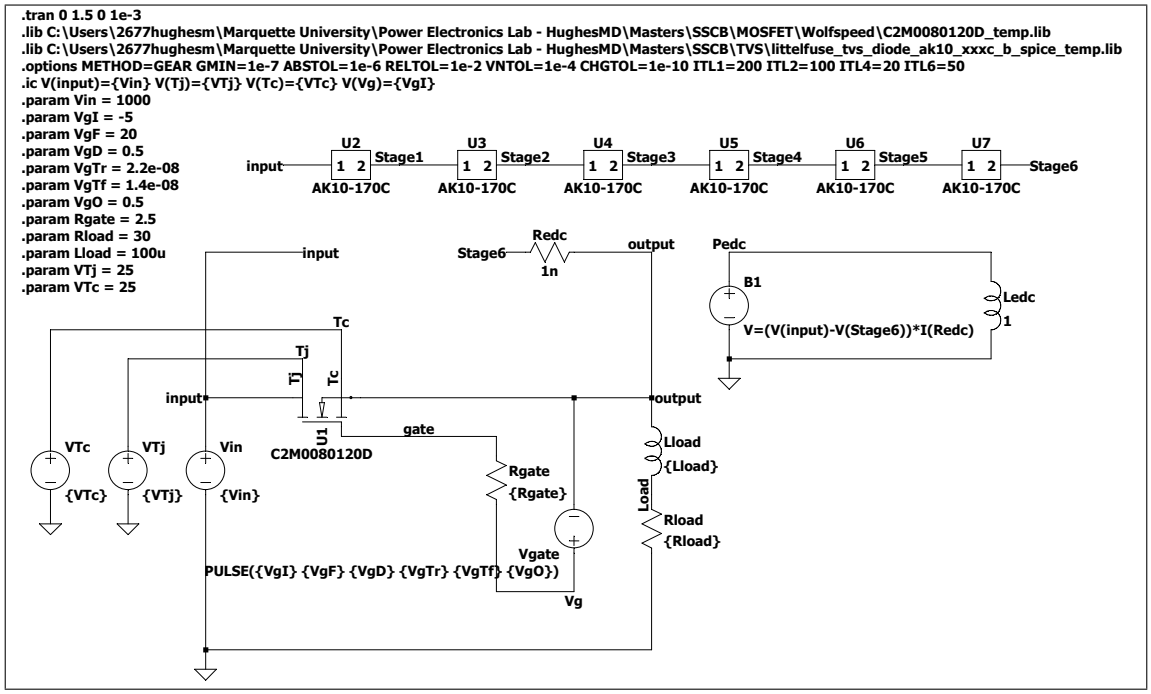


Figure B.32: TVS Diode Unidirectional Test Circuit of TVS 1 using Device 8

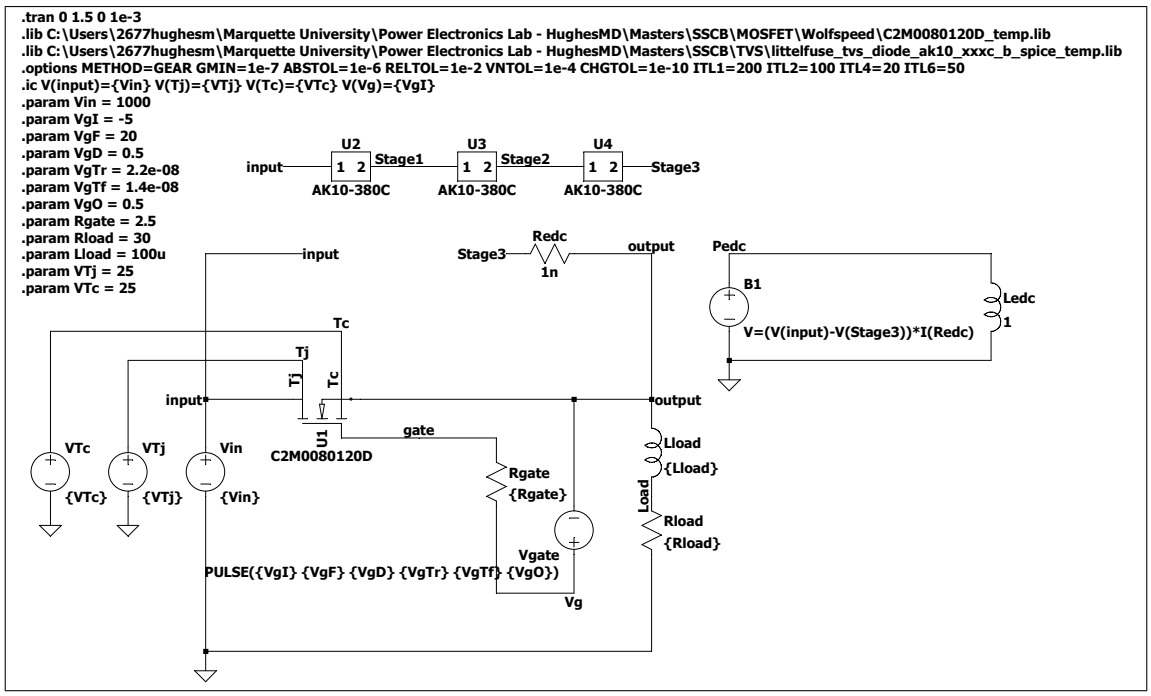


Figure B.33: TVS Diode Unidirectional Test Circuit of TVS 2 using Device 8

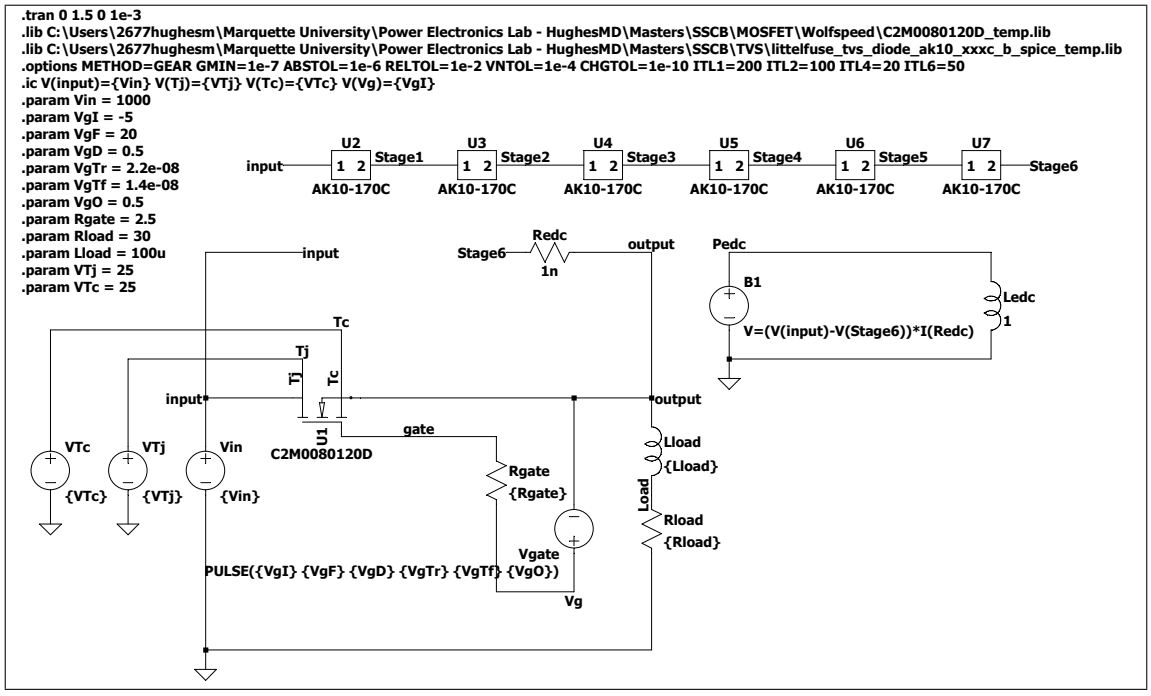


Figure B.34: TVS Diode Unidirectional Test Circuit of TVS 3 using Device 8

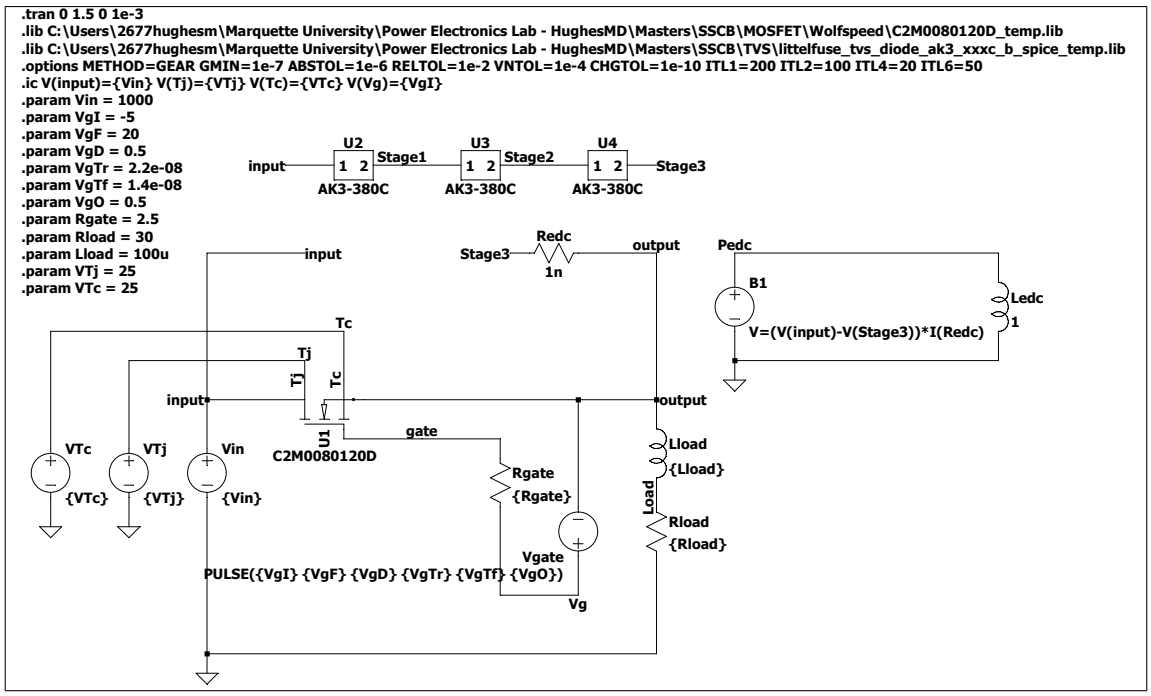


Figure B.35: TVS Diode Unidirectional Test Circuit of TVS 4 using Device 8

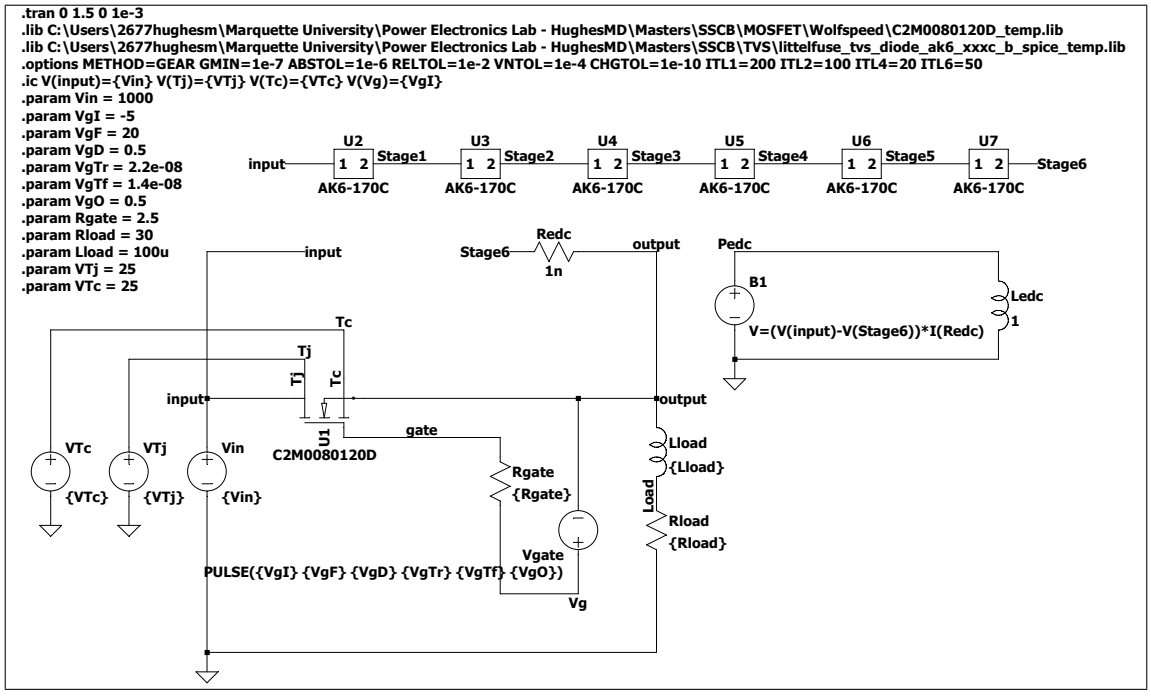


Figure B.36: TVS Diode Unidirectional Test Circuit of TVS 5 using Device 8

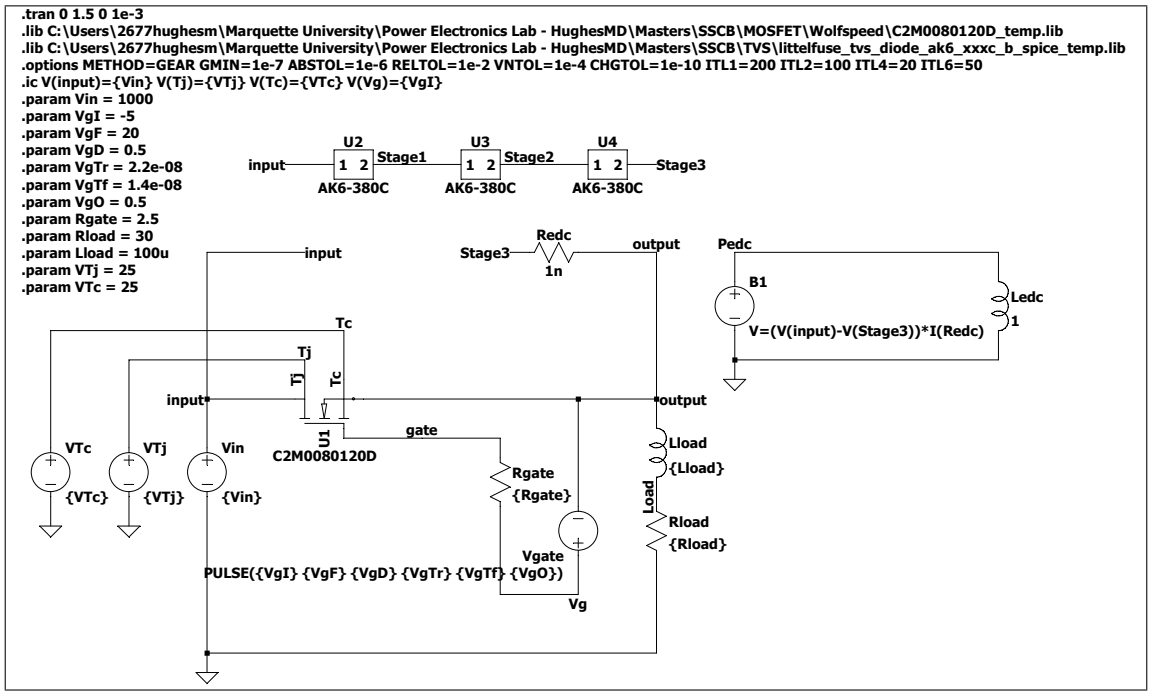


Figure B.37: TVS Diode Unidirectional Test Circuit of TVS 6 using Device 8

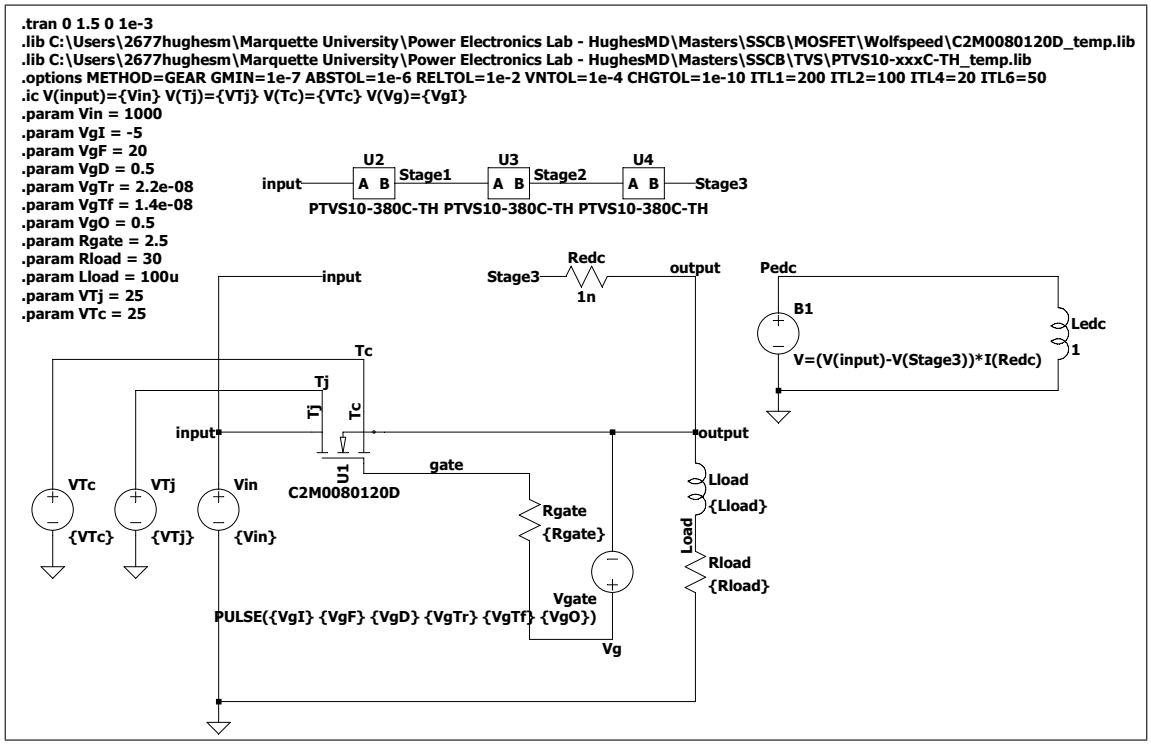


Figure B.38: TVS Diode Unidirectional Test Circuit of TVS 7 using Device 8

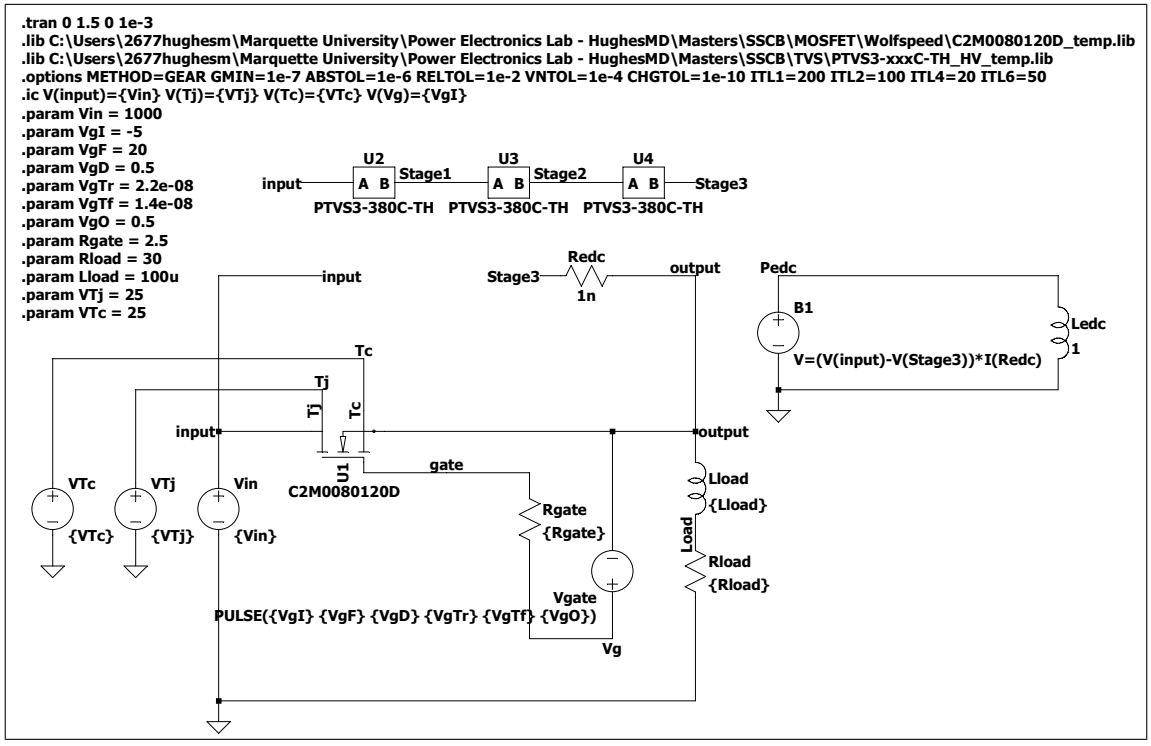


Figure B.39: TVS Diode Unidirectional Test Circuit of TVS 8 using Device 8

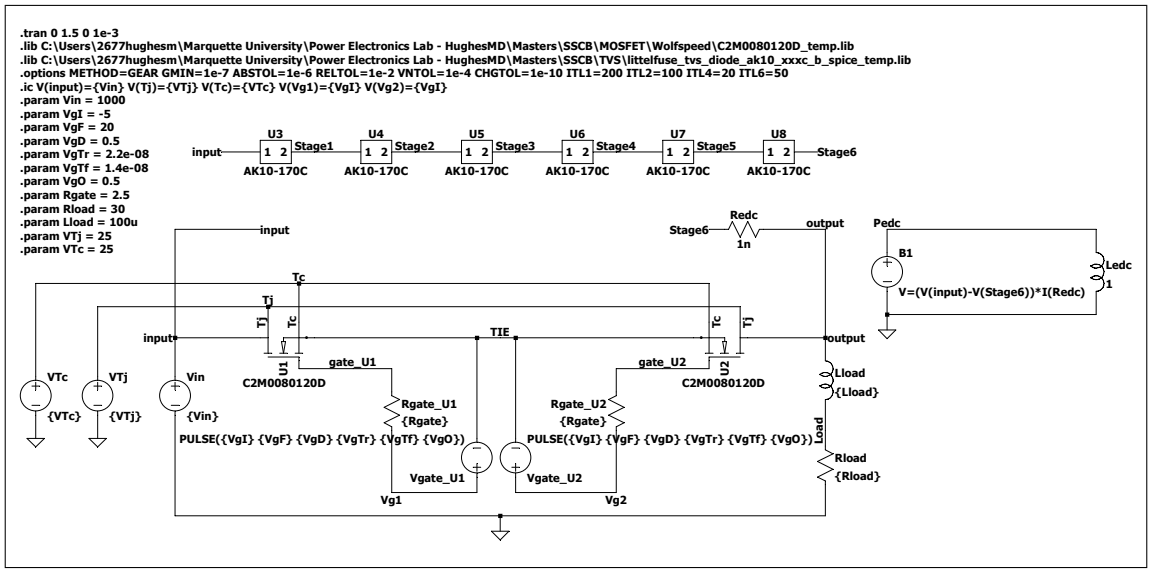


Figure B.40: TVS Diode Bidirectional Test Circuit of TVS 1 using Device 8

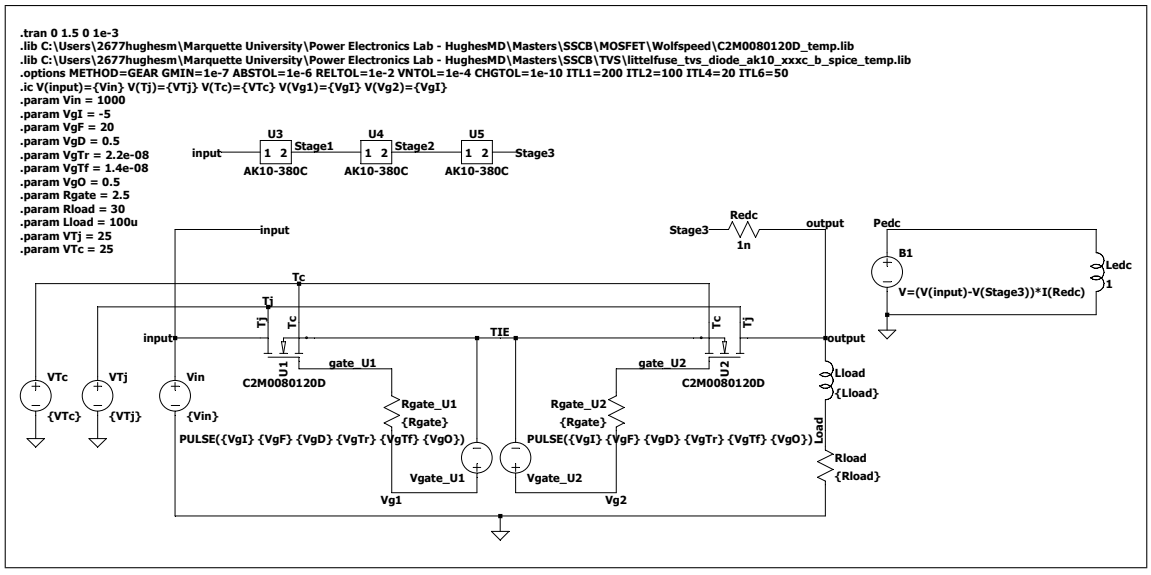


Figure B.41: TVS Diode Bidirectional Test Circuit of TVS 2 using Device 8

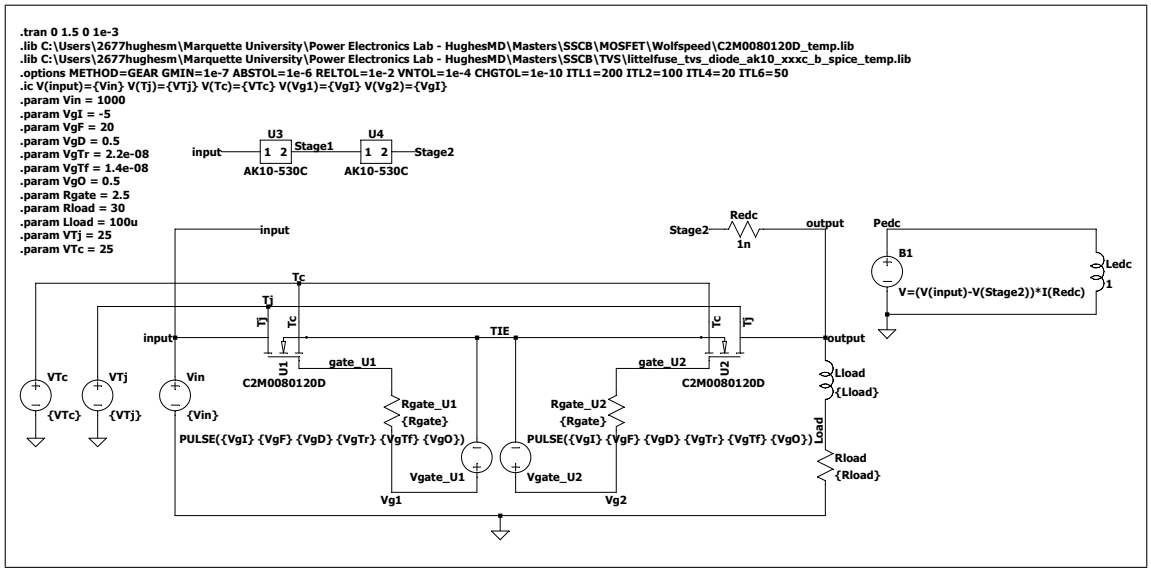


Figure B.42: TVS Diode Bidirectional Test Circuit of TVS 3 using Device 8

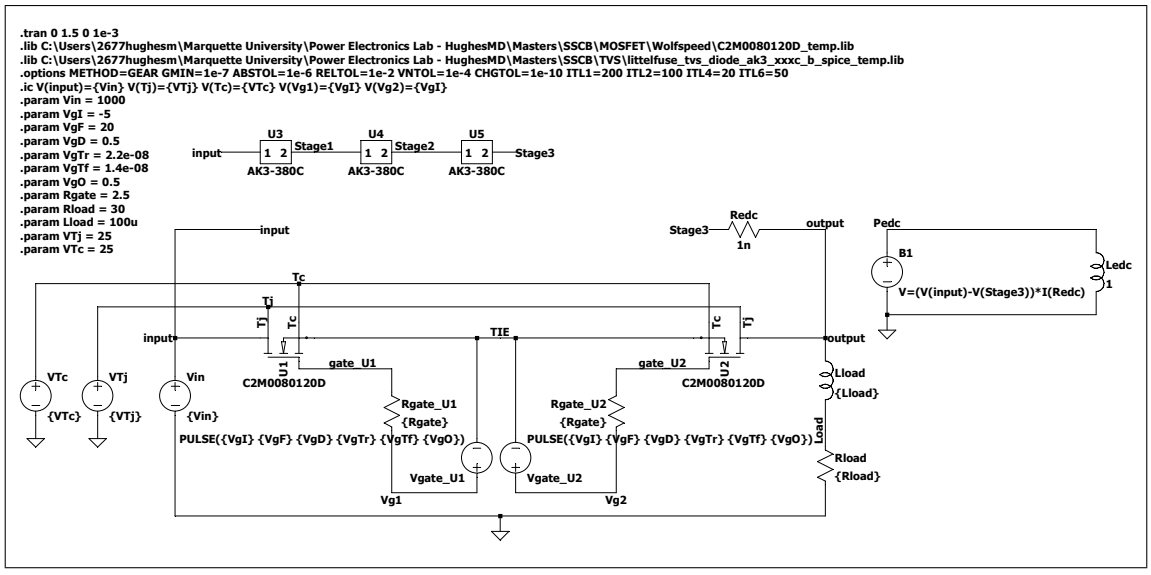


Figure B.43: TVS Diode Bidirectional Test Circuit of TVS 4 using Device 8

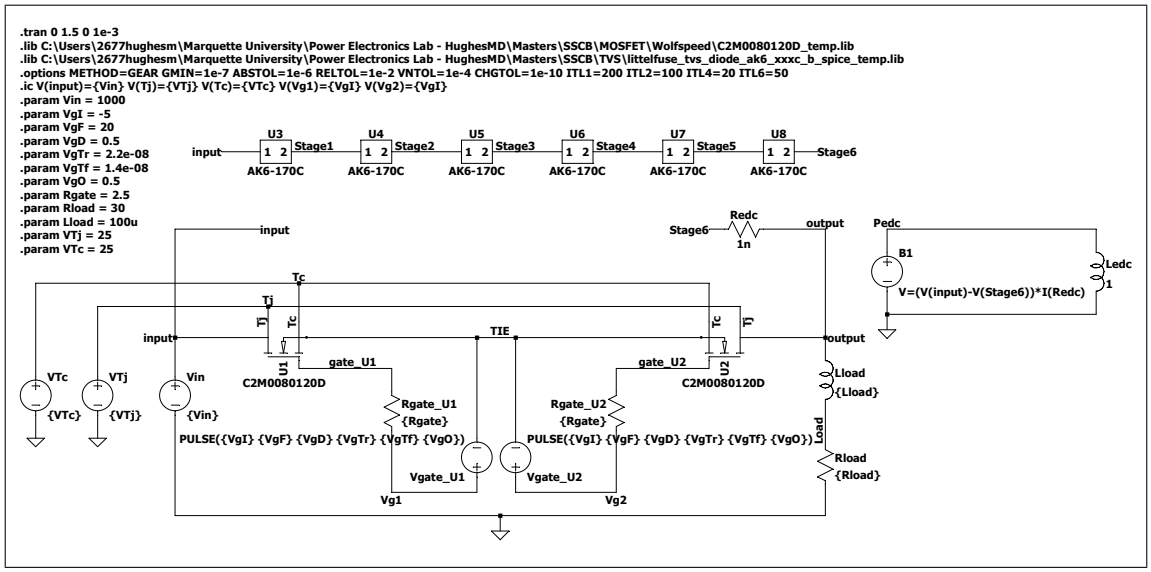


Figure B.44: TVS Diode Bidirectional Test Circuit of TVS 5 using Device 8

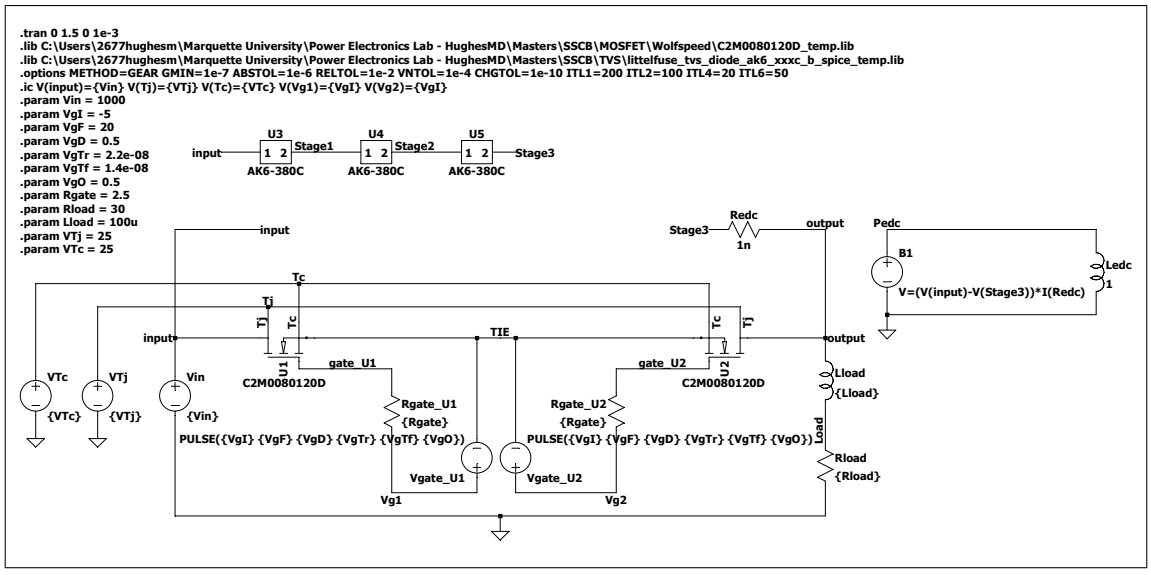


Figure B.45: TVS Diode Bidirectional Test Circuit of TVS 6 using Device 8

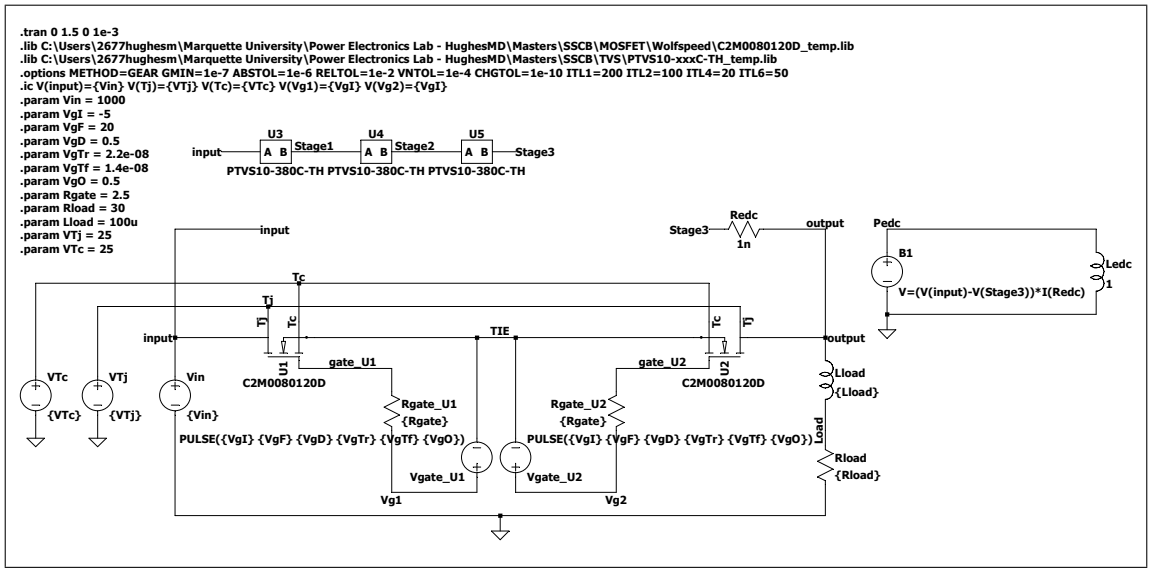


Figure B.46: TVS Diode Bidirectional Test Circuit of TVS 7 using Device 8

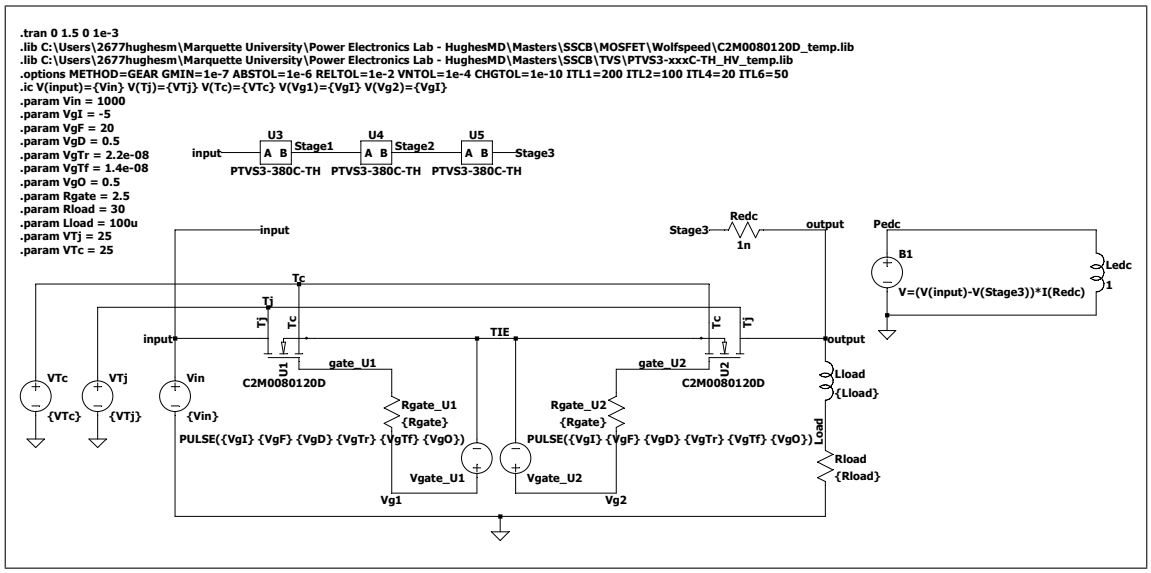


Figure B.47: TVS Diode Bidirectional Test Circuit of TVS 8 using Device 8

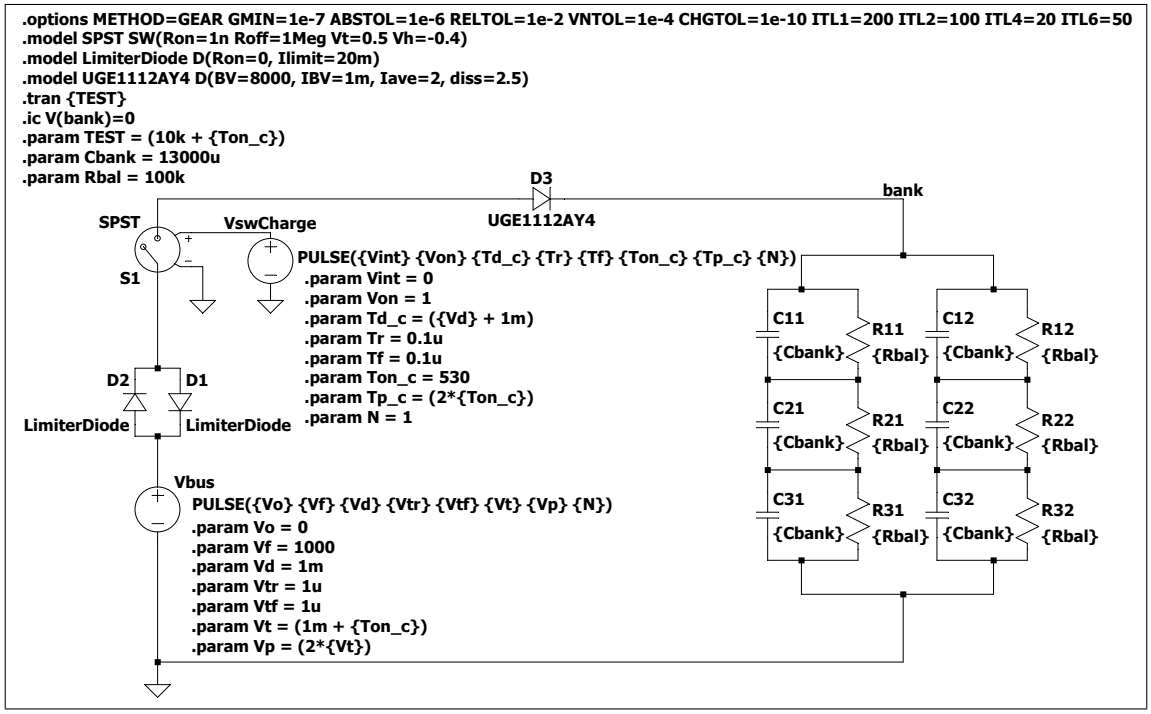


Figure B.48: SSCB Short Circuit Test Fixture Charging Sequence Simulation

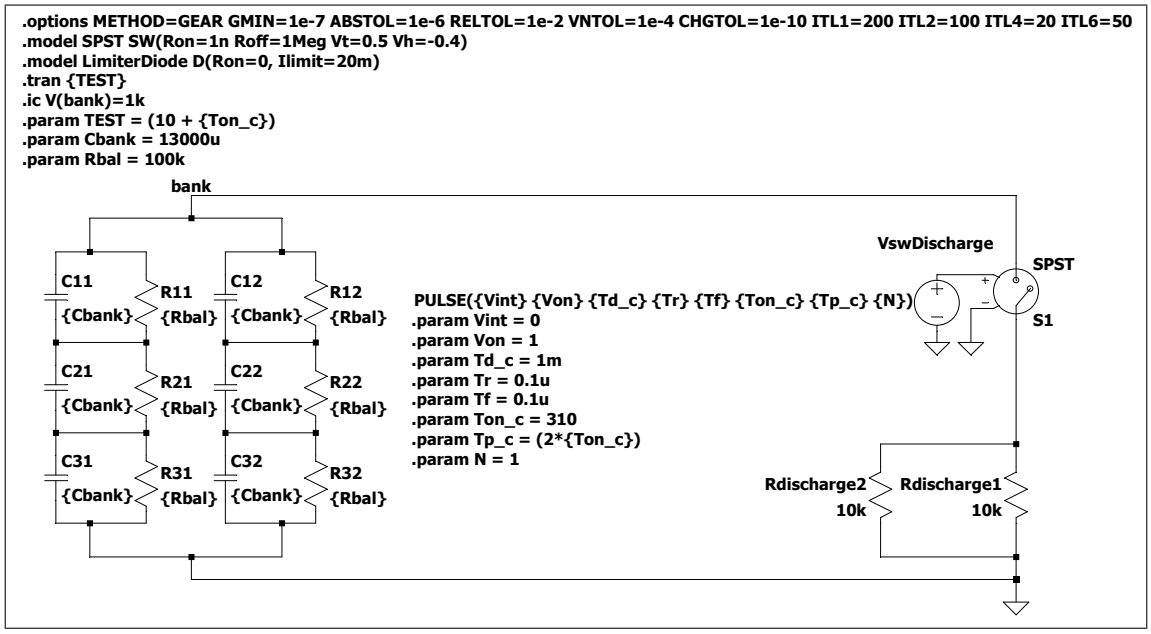


Figure B.49: SSCB Short Circuit Test Fixture Discharging Sequence Simulation

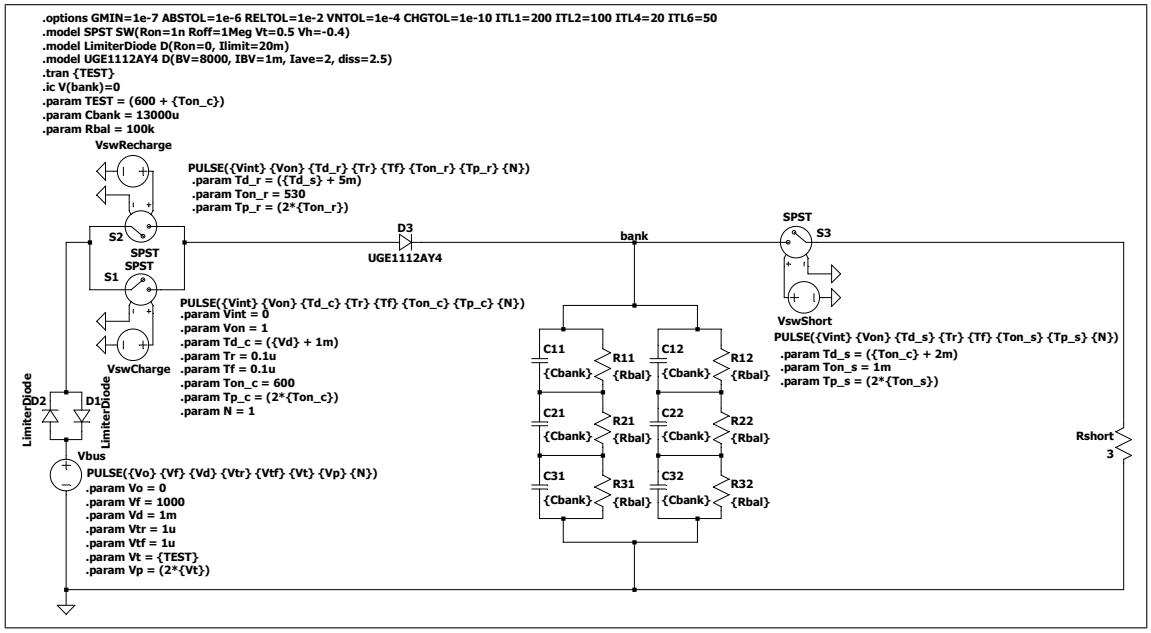


Figure B.50: SSCB Short Circuit Test Fixture Recharging Sequence Simulation

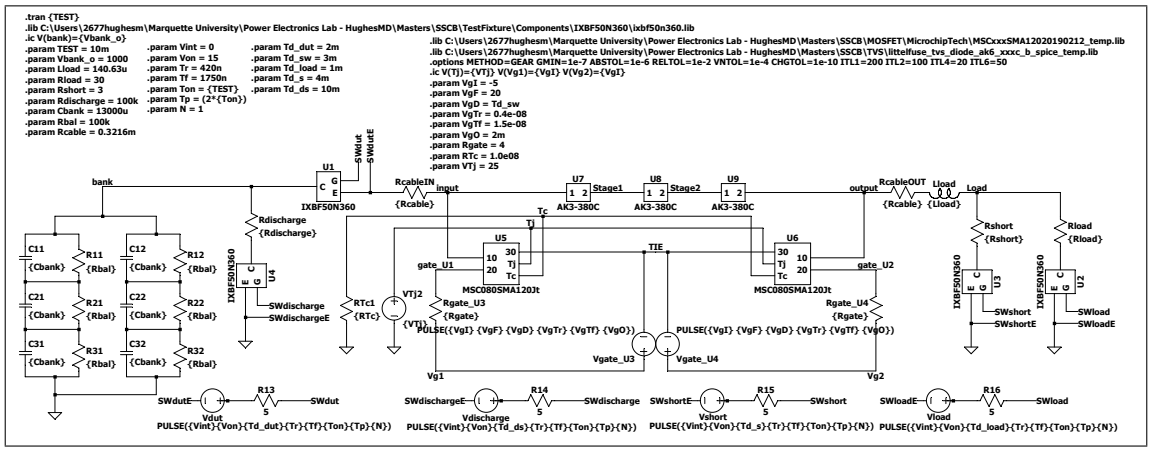


Figure B.51: SSCB Short Circuit Test Fixture Simulation

APPENDIX C
SOFTWARE CODE

```
%% SSCB MOSFET Efficiency
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Infineon\IMW120R060M1H\SwitchEfficiency\SSCB_uni_Inf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Infineon\CoolSiC_MOSFET_1200V_G1_30-350mOhm_1.1_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\MOSFET\Infineon\']; %data file path
circuit_filepath = [data_filepath 'IMW120R060M1H\SwitchEfficiency\SSCB_uni_Inf']; %circuit simulation file path
component_filepath = [data_filepath 'CoolSiC_MOSFET_1200V_G1_30-350mOhm_1.1_temp.lib']; %switch model file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -0.1; %initial gate voltage
VgateF_vec = 18; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 17 *1e-9; %rise time
VgateTf_vec = 12 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 2; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_uni_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
```

```
48 %           name            node1  node2  node3  node4  node5    value /
       file with values
49 line{02} = 'Vin             input  0                             {Vin}';
50 line{03} = 'XU1             input  gate   output Tj     Tc
       IMW120R060M1H_L3';
51 line{04} = 'Rgate           gate   Vg                            {Rgate}';
52 line{05} = 'Vgate           Vg     output                        PULSE({VgI}
        {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet, Tr is
       middle of 7n 17n
53 line{06} = 'Rload           output 0                             {Rload}';
54 line{07} = 'VTj             Tj     0                             {VTj}';
55 line{08} = 'VTc             Tc     0                             {VTc}';
56 line{09} = 'C1              Tj     0                             1pf';
57 line{10} = 'C2              Tc     0                             1pf';
58
59 line{11} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop) ' ' num2str(Tstart)
       ' ' num2str(Tmaxstep)];
60 line{12} = ['.lib ' component_filepath];
61 line{13} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
       -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
62 line{14} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg)={VgI}';
63 line{15} = ['.param Vin = ' num2str(Vin_vec)];
64 line{16} = ['.param VgI = ' num2str(VgateI_vec)];
65 line{17} = ['.param VgF = ' num2str(VgateF_vec)];
66 line{18} = ['.param VgD = ' num2str(VgateD_vec)];
67 line{19} = ['.param VgTr = ' num2str(VgateTr_vec)];
68 line{20} = ['.param VgTf = ' num2str(VgateTf_vec)];
69 line{21} = ['.param VgO = ' num2str(VgateO_vec)];
70 line{22} = ['.param Rgate = ' num2str(Rgate_vec)];
71 line{23} = ['.step param Rload ' Rload_vec];
72 line{24} = ['.step param VTj list ' num2str(Tj_vec)];
73 line{25} = ['.param VTc = ' num2str(Tc_vec)];
74 line{26} = '.backanno';
75 line{27} = '.end';
76
77 %% New Netlist
78 fid = fopen([circuit_filepath '.cir'],'wb');
79 for i = 1:length(line)
80     fwrite(fid, [line{i} char(13) newline], 'char');
81 end
82 fid = fclose(fid);
83
84 %% Simulate
85 system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
       circuit_filepath '.cir"']);
86
87 %% Data Collection
88 warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
       encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
        In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
89 raw_data = LTspice2Matlab([circuit_filepath '.raw']);
90
91 %UniDirectional Circuit
92 time = raw_data.time_vect;
93 for ii = 1:size(time,2)
94     if time(ii) >= AVGstart
95         target(span) = ii;
96         span = span+1;
97         if time(ii) == Tstop
```

```
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(14,target));
            Pin = Vin.*abs(Iin);

            Vd = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs = nanmean(raw_data.variable_mat(3,target)); %same as Vout
            Vds = Vd-Vs;
            Pds = Vds.*abs(Iin);

            Efficiency = 100 * (Pin-Pds)./Pin;

            Tj = nanmean(raw_data.variable_mat(4,target)); %per doc VTj = Tj

            SWeff_uni_mat(Row,1) = abs(Iin);
            SWeff_uni_mat(Row,2) = Tj;
            SWeff_uni_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

%% Data Save
save([data_filepath 'IMW120R060M1H\SwitchEfficiency\SWeff_uni_mat_temp.mat'
    ],'SWeff_uni_mat');
```

Software C.1: Efficiency Simulation of a Unidirectional circuit using an Device 1

```
%% SSCB MOSFET Efficiency
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Littelfuse\LSIC1MO120E0080\SwitchEfficiency\
    SSCB_uni_Littelfuse.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Littelfuse\LSIC1MO120E0080\LSIC1MO120E0080_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\MOSFET\Littelfuse\']; %data file path
circuit_filepath = [data_filepath 'LSIC1MO120E0080\SwitchEfficiency\
    SSCB_uni_Littelfuse']; %circuit simulation file path
component_filepath = [data_filepath 'LSIC1MO120E0080_temp.lib']; %switch
    model file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging
```

```
23
24 Vin_vec = 1000; %input voltage
25
26 VgateI_vec = -5; %initial gate voltage
27 VgateF_vec = 20; %final gate voltage
28 VgateD_vec = 0.5; %gate voltage delay
29 VgateTr_vec = 10 *1e-9; %rise time
30 VgateTf_vec = 6 *1e-9; %fall time
31 VgateO_vec = 0.5; %gate voltage ON time
32
33 Rgate_vec = 2; %settings from datasheet
34
35 Rload_vec = '30 100 1'; %sets load resistance row vector
36
37 Tj_vec = [-55 25 150 175]; %sets Tj row vector
38 Tc_vec = 25; %sets Tc row vector
39
40 span = 1; %initialize average start
41 target = zeros(1,1); %initializes target time matrix
42 SWeff_uni_mat = zeros(284,3); %initializes switch efficiency matrix
43 Row = 1; %intitialize row count for data dump
44
45 %% MatLab Netlist
46 line{01} = ['* ' circuit_filepath '.asc'];
47 %UniDirectional Circuit
48 %           name            node1  node2  node3  node4  node5    value /
    file with values
49 line{02} = 'Vin             input  0                              {Vin}';
50 line{03} = 'XU1             input  gate   output
    LSIC1MO120E0080 T = {Tj}';
51 line{04} = 'Rgate           gate   Vg                             {Rgate}';
52 line{05} = 'Vgate           Vg     output                         PULSE({VgI}
     {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet are -5/20
53 line{06} = 'Rload           output 0                              {Rload}';
54 line{07} = 'VTj             Tj     0                              {Tj}';
55
56 line{08} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep)];
57 line{09} = ['.lib ' component_filepath];
58 line{10} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
59 line{11} = '.ic V(input)={Vin} V(Vg)={VgI}';
60 line{12} = ['.param Vin = ' num2str(Vin_vec)];
61 line{13} = ['.param VgI = ' num2str(VgateI_vec)];
62 line{14} = ['.param VgF = ' num2str(VgateF_vec)];
63 line{15} = ['.param VgD = ' num2str(VgateD_vec)];
64 line{16} = ['.param VgTr = ' num2str(VgateTr_vec)];
65 line{17} = ['.param VgTf = ' num2str(VgateTf_vec)];
66 line{18} = ['.param VgO = ' num2str(VgateO_vec)];
67 line{19} = ['.param Rgate = ' num2str(Rgate_vec)];
68 line{20} = ['.step param Rload ' Rload_vec];
69 line{21} = ['.step param Tj list ' num2str(Tj_vec)];
70 line{22} = '.backanno';
71 line{23} = '.end';
72
73 %% New Netlist
74 fid = fopen([circuit_filepath '.cir'],'wb');
75 for i = 1:length(line)
76     fwrite(fid, [line{i} char(13) newline], 'char');
```

```
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(10,target));
            Pin = Vin.*abs(Iin);

            Vd = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs = nanmean(raw_data.variable_mat(2,target)); %same as Vout
            Vds = Vd-Vs;
            Pds = Vds.*abs(Iin);

            Efficiency = 100 * (Pin-Pds)./Pin;

            Tj = nanmean(raw_data.variable_mat(5,target)); %per doc VTj = Tj

            SWeff_uni_mat(Row,1) = abs(Iin);
            SWeff_uni_mat(Row,2) = Tj;
            SWeff_uni_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

%% Data Save
save([data_filepath 'LSIC1MO120E0080\SwitchEfficiency\SWeff_uni_mat_temp.mat
    '],'SWeff_uni_mat');
```

Software C.2: Efficiency Simulation of a Unidirectional circuit using Device 2

```
%% SSCB MOSFET Efficiency
clc
close all
clear variables

%% Paths
```

```
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\MicrochipTech\MSC080SMA120J\SwitchEfficiency\SSCB_uni_MicroT.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\MicrochipTech\MSC080SMA120J.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\MOSFET\MicrochipTech\']; %data file path
circuit_filepath = [data_filepath 'MSC080SMA120J\SwitchEfficiency\SSCB_uni_MicroT']; %circuit simulation file path
component_filepath = [data_filepath 'MSCxxxSMA12020190212_temp.lib']; %switch model file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 4 *1e-9; %rise time
VgateTf_vec = 15 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 4; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 100 *1e6; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_uni_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value / file with values
line{02} = 'Vin             input  0                             {Vin}';
line{03} = 'XU1             input  gate   output Tj     Tc       MSC080SMA120Jt';
line{04} = 'Rgate           gate   Vg                            {Rgate}';
line{05} = 'Vgate           Vg     output                        PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{06} = 'Rload           output 0                             {Rload}';
line{07} = 'VTj             Tj     0                             {VTj}';
line{08} = 'RTc             Tc     0                             {RTc}';

```

```
line{09} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart) ' ' num2str(Tmaxstep)];
line{10} = ['.lib ' component_filepath];
line{11} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e-4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{12} = '.ic V(input)={Vin} V(Tj)={VTj} V(Vg)={VgI}';
line{13} = ['.param Vin = ' num2str(Vin_vec)];
line{14} = ['.param VgI = ' num2str(VgateI_vec)];
line{15} = ['.param VgF = ' num2str(VgateF_vec)];
line{16} = ['.param VgD = ' num2str(VgateD_vec)];
line{17} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{18} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{19} = ['.param VgO = ' num2str(VgateO_vec)];
line{20} = ['.param Rgate = ' num2str(Rgate_vec)];
line{21} = ['.step param Rload ' Rload_vec];
line{22} = ['.step param VTj list ' num2str(Tj_vec)];
line{23} = ['.param RTc = ' num2str(Tc_vec)];
line{24} = '.backanno';
line{25} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "' circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. > In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(12,target));
            Pin = Vin.*abs(Iin);

            Vd = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs = nanmean(raw_data.variable_mat(3,target)); %same as Vout
            Vds = Vd-Vs;
            Pds = Vds.*abs(Iin);

            Efficiency = 100 * (Pin-Pds)./Pin;

            Tj = nanmean(raw_data.variable_mat(4,target)); %per doc VTj = Tj

            SWeff_uni_mat(Row,1) = abs(Iin);
            SWeff_uni_mat(Row,2) = Tj;
```

```
            SWeff_uni_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

%% Data Save
save([data_filepath 'MSC080SMA120J\SwitchEfficiency\SWeff_uni_mat_temp.mat'
    ],'SWeff_uni_mat');
```

Software C.3: Efficiency Simulation of a Unidirectional circuit using Device 3

```
%% SSCB MOSFET Efficiency
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Rohm\SCT3080KLGC11\SwitchEfficiency\SSCB_uni_Rohm.
    cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Rohm\SCT3080KL_LT_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\MOSFET\Rohm\']; %data file path
circuit_filepath = [data_filepath 'SCT3080KLGC11\SwitchEfficiency\
    SSCB_uni_Rohm']; %circuit simulation file path
component_filepath = [data_filepath 'SCT3080KL_LT_temp.lib']; %switch model
    file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = 0; %initial gate voltage
VgateF_vec = 18; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 15 *1e-9; %rise time
VgateTf_vec = 24 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

```

```
span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_uni_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

T0_vec = Tj_vec; %sets Tj

%% Temperature Modification
for iii = 1:size(T0_vec,2)

T0 = T0_vec(iii); %sets Tj

%% Matlab Model File
fid = fopen(component_filepath);
lines = textscan(fid,'%s','delimiter','\n');
fclose(fid);
lines = lines{1};

%modified lines
lines{10,1} = ['.PARAM T0=' num2str(T0)];

fid = fopen(component_filepath,'w');
fprintf(fid,'%s\n',lines{:});
fclose(fid);

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value /
    file with values
line{02} = 'Vin             input  0                              {Vin}';
line{03} = 'XU1             input  Vg     output
    SCT3080KL_LT';
line{04} = 'Vgate           Vg     output                         PULSE({VgI}
     {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{05} = 'Rload           output 0                              {Rload}';

line{06} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep)];
line{07} = ['.lib ' component_filepath];
line{08} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{09} = '.ic V(input)={Vin} V(Vg)={VgI}';
line{10} = ['.param Vin = ' num2str(Vin_vec)];
line{11} = ['.param VgI = ' num2str(VgateI_vec)];
line{12} = ['.param VgF = ' num2str(VgateF_vec)];
line{13} = ['.param VgD = ' num2str(VgateD_vec)];
line{14} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{15} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{16} = ['.param VgO = ' num2str(VgateO_vec)];
line{17} = ['.step param Rload ' Rload_vec];
line{18} = '.backanno';
line{19} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
```

```
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(6,target));
            Pin = Vin.*abs(Iin);

            Vd = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs = nanmean(raw_data.variable_mat(3,target)); %same as Vout
            Vds = Vd-Vs;
            Pds = Vds.*abs(Iin);

            Efficiency = 100 * (Pin-Pds)./Pin;

            Tj = T0;

            SWeff_uni_mat(Row,1) = abs(Iin);
            SWeff_uni_mat(Row,2) = Tj;
            SWeff_uni_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

end %ends Tj for loop

%% Data Save
save([data_filepath 'SCT3080KLGC11\SwitchEfficiency\SWeff_uni_mat_temp.mat'
    ],'SWeff_uni_mat');
```

Software C.4: Efficiency Simulation of a Unidirectional circuit using Device 4

```
%% SSCB JFET Efficiency
clc
close all
clear variables

%% Paths
```

```
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\JFET\UnitedSiC\UJ3N120070K3S\SwitchEfficiency\SSCB_uni_USiC.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\JFET\UnitedSiC\UJ3N120070K3S_temp.301
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\JFET\UnitedSiC\']; %data file path
circuit_filepath = [data_filepath 'UJ3N120070K3S\SwitchEfficiency\SSCB_uni_USiC']; %circuit simulation file path
component_filepath = [data_filepath 'UJ3N120070K3S_temp.301']; %switch model file path

load([data_filepath 'Tj_Rds_on_USiC.mat']); %temp curve for switch

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -18; %initial gate voltage
VgateF_vec = 0; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 25 *1e-9; %rise time
VgateTf_vec = 39 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 1; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [1 81 206 231]; %sets Tj row vector, corresponds to -55, 25, 150, 175
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_uni_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% Rds_on Modification
for iii = 1:size(Tj_vec,2)

Tj_row = Tj_vec(iii); %sets Tj
Rds_on = Tj_Rds_on_USiC(Tj_row,3); %sets Rds_on
Tj = Tj_Rds_on_USiC(Tj_row,1); %sets Tj

%% Matlab Model File
fid = fopen(component_filepath);
lines = textscan(fid,'%s','delimiter','\n');
fclose(fid);
lines = lines{1};
```

```
59
60 %modified lines
61 lines{19,1} = 'Ld nd nd1 5n'; %modified to maintain spacing, tab
       incidentially deleted on input
62 lines{20,1} = 'Ljg ng ng1 10n'; %modified to maintain spacing, tab
       incidentially deleted on input
63 lines{21,1} = 'Ls ns ns1 2n'; %modified to maintain spacing, tab
       incidentially deleted on input
64 lines{22,1} = ['xj1 nd1 ng1 ns1 jfet_G3_1200V_Ron params: Ron=' num2str(
       Rds_on) ' Rgoff=3.5 Rgon=3.5'];
65
66 fid = fopen(component_filepath,'w');
67 fprintf(fid,'%s\n',lines{:});
68 fclose(fid);
69
70 %% MatLab Netlist
71 line{01} = ['* ' circuit_filepath '.asc'];
72 %UniDirectional Circuit
73 %           name            node1  node2  node3  node4  node5    value /
       file with values
74 line{02} = 'Vin             input  0                              {Vin}';
75 line{03} = 'XU1             input  gate   output
       UJ3N120070K3S';
76 line{04} = 'Rgate           gate   Vg                             {Rgate}';
77 line{05} = 'Vgate           Vg     output                         PULSE({VgI}
        {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
78 line{06} = 'Rload           output 0                              {Rload}';
79
80 line{07} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
        ' ' num2str(Tmaxstep)];
81 line{08} = ['.lib ' component_filepath];
82 line{09} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
       -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
83 line{10} = '.ic V(input)={Vin} V(Vg)={VgI}';
84 line{11} = ['.param Vin = ' num2str(Vin_vec)];
85 line{12} = ['.param VgI = ' num2str(VgateI_vec)];
86 line{13} = ['.param VgF = ' num2str(VgateF_vec)];
87 line{14} = ['.param VgD = ' num2str(VgateD_vec)];
88 line{15} = ['.param VgTr = ' num2str(VgateTr_vec)];
89 line{16} = ['.param VgTf = ' num2str(VgateTf_vec)];
90 line{17} = ['.param VgO = ' num2str(VgateO_vec)];
91 line{18} = ['.param Rgate = ' num2str(Rgate_vec)];
92 line{19} = ['.step param Rload ' Rload_vec];
93 line{20} = '.backanno';
94 line{21} = '.end';
95
96 %% New Netlist
97 fid = fopen([circuit_filepath '.cir'],'wb');
98 for i = 1:length(line)
99     fwrite(fid, [line{i} char(13) newline], 'char');
100 end
101 fid = fclose(fid);
102
103 %% Simulate
104 system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
       circuit_filepath '.cir"']);
105
106 %% Data Collection
```

```
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. > In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(8,target));
            Pin = Vin.*abs(Iin);

            Vd = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs = nanmean(raw_data.variable_mat(3,target)); %same as Vout
            Vds = Vd-Vs;
            Pds = Vds.*abs(Iin);

            Efficiency = 100 * (Pin-Pds)./Pin;

            Tj = Tj;

            SWeff_uni_mat(Row,1) = abs(Iin);
            SWeff_uni_mat(Row,2) = Tj;
            SWeff_uni_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

end %ends Rds_on for loop

%% Data Save
save([data_filepath 'UJ3N120070K3S\SwitchEfficiency\SWeff_uni_mat_temp.mat'],'SWeff_uni_mat');
```

Software C.5: Efficiency Simulation of a Unidirectional circuit using an Device 5

```
%% SSCB JFET Efficiency
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\JFET\UnitedSiC\UJ3C120080K3S\SwitchEfficiency\SSCB_uni_USiC.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\JFET\UnitedSiC\UJ3C120080K3S_temp.301
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
```

```
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\JFET\UnitedSiC\']; %data file path
circuit_filepath = [data_filepath 'UJ3C120080K3S\SwitchEfficiency\SSCB_uni_USiC']; %circuit simulation file path
component_filepath = [data_filepath 'UJ3C120080K3S_temp.301']; %switch model file path

load([data_filepath 'Tj_Rds_on_USiC.mat']); %temp curve for switch

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 15; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 14 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 1; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [1 81 206 231]; %sets Tj row vector, corresponds to -55, 25, 150, 175
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_uni_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% Rds_on Modification
for iii = 1:size(Tj_vec,2)

Tj_row = Tj_vec(iii); %sets Tj
Rds_on = Tj_Rds_on_USiC(Tj_row,3); %sets Rds_on
Tj = Tj_Rds_on_USiC(Tj_row,1); %sets Tj

%% Matlab Model File
fid = fopen(component_filepath);
lines = textscan(fid,'%s','delimiter','\n');
fclose(fid);
lines = lines{1};

%modified lines
lines{18,1} = 'Ld nd nd1 5n'; %modified to maintain spacing, tab incidentially deleted on input
lines{19,1} = 'Lmd ns1 nd2 2n';
lines{20,1} = 'Ljg ng1 ns3 4n'; %modified to maintain spacing, tab incidentially deleted on input
```

```
lines{21,1} = 'Lmg ng ng2 70n'; %modified to maintain spacing, tab incidentially deleted on input
lines{22,1} = 'Lms ns2 ns3 3n';
lines{23,1} = 'Ls ns3 ns 2n';
lines{24,1} = ['xj1 nd1 ng1 ns1 jfet_G3_1200V_Ron params: Ron=' num2str(Rds_on) ' Rgoff=1.7 Rgon=1.7'];
lines{25,1} = 'xm1 nd2 ng2 ns2 mfet180';

fid = fopen(component_filepath,'w');
fprintf(fid,'%s\n',lines{:});
fclose(fid);

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value / file with values
line{02} = 'Vin             input  0                             {Vin}';
line{03} = 'XU1             input  gate   output                 UJ3C120080K3S';
line{04} = 'Rgate           gate   Vg                            {Rgate}';
line{05} = 'Vgate           Vg     output                        PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{06} = 'Rload           output 0                             {Rload}';

line{07} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart) ' ' num2str(Tmaxstep)];
line{08} = ['.lib ' component_filepath];
line{09} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e-4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{10} = '.ic V(input)={Vin} V(Vg)={VgI}';
line{11} = ['.param Vin = ' num2str(Vin_vec)];
line{12} = ['.param VgI = ' num2str(VgateI_vec)];
line{13} = ['.param VgF = ' num2str(VgateF_vec)];
line{14} = ['.param VgD = ' num2str(VgateD_vec)];
line{15} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{16} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{17} = ['.param VgO = ' num2str(VgateO_vec)];
line{18} = ['.param Rgate = ' num2str(Rgate_vec)];
line{19} = ['.step param Rload ' Rload_vec];
line{20} = '.backanno';
line{21} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "' circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. > In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);
```

```
113
114 %UniDirectional Circuit
115 time = raw_data.time_vect;
116 for ii = 1:size(time,2)
117     if time(ii) >= AVGstart
118         target(span) = ii;
119         span = span+1;
120         if time(ii) == Tstop
121             Vin = nanmean(raw_data.variable_mat(1,target));
122             Iin = nanmean(raw_data.variable_mat(8,target));
123             Pin = Vin.*abs(Iin);
124
125             Vd = nanmean(raw_data.variable_mat(1,target)); %same as Vin
126             Vs = nanmean(raw_data.variable_mat(3,target)); %same as Vout
127             Vds = Vd-Vs;
128             Pds = Vds.*abs(Iin);
129
130             Efficiency = 100 * (Pin-Pds)./Pin;
131
132             Tj = Tj;
133
134             SWeff_uni_mat(Row,1) = abs(Iin);
135             SWeff_uni_mat(Row,2) = Tj;
136             SWeff_uni_mat(Row,3) = Efficiency;
137
138             Row = Row+1; %increments to next row for data dump
139             clear target
140             span = 1;
141         end
142     end
143 end
144
145 end %ends Rds_on for loop
146
147 %% Data Save
148 save([data_filepath 'UJ3C120080K3S\SwitchEfficiency\SWeff_uni_mat_temp.mat'
    ],'SWeff_uni_mat');
```

Software C.6: Efficiency Simulation of a Unidirectional circuit using an Device 6

```
1 %% SSCB JFET Efficiency
2 clc
3 close all
4 clear variables
5
6 %% Paths
7 %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\JFET\UnitedSiC\UJ3C120070K3S\SwitchEfficiency\SSCB_uni_USiC
    .cir
8 %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\JFET\UnitedSiC\UJ3C120070K3S_temp.301
9 %C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe
10
11 %% Setup
12 parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
13 data_filepath = [parent_filepath '\JFET\UnitedSiC\']; %data file path
```

```
14 circuit_filepath = [data_filepath 'UJ3C120070K3S\SwitchEfficiency\
       SSCB_uni_USiC']; %circuit simulation file path
15 component_filepath = [data_filepath 'UJ3C120070K3S_temp.301']; %switch model
       file path
16
17 load([data_filepath 'Tj_Rds_on_USiC.mat']); %temp curve for switch
18
19 %all parameters are scriptable, use .step param or .step param XXX list
20 Tprint = 0;
21 Tstop = 1; %transient analysis stop time
22 Tstart = 0;
23 Tmaxstep = 1 *1e-3; %transient analysis step
24 AVGstart = 0.9; %start averaging
25
26 Vin_vec = 1000; %input voltage
27
28 VgateI_vec = -5; %initial gate voltage
29 VgateF_vec = 15; %final gate voltage
30 VgateD_vec = 0.5; %gate voltage delay
31 VgateTr_vec = 17 *1e-9; %rise time
32 VgateTf_vec = 9 *1e-9; %fall time
33 VgateO_vec = 0.5; %gate voltage ON time
34
35 Rgate_vec = 1; %settings from datasheet
36
37 Rload_vec = '30 100 1'; %sets load resistance row vector
38
39 Tj_vec = [1 81 206 231]; %sets Tj row vector, corresponds to -55, 25, 150,
       175
40 Tc_vec = 25; %sets Tc row vector
41
42 span = 1; %initialize average start
43 target = zeros(1,1); %initializes target time matrix
44 SWeff_uni_mat = zeros(284,3); %initializes switch efficiency matrix
45 Row = 1; %intitialize row count for data dump
46
47 %% Rds_on Modification
48 for iii = 1:size(Tj_vec,2)
49
50 Tj_row = Tj_vec(iii); %sets Tj
51 Rds_on = Tj_Rds_on_USiC(Tj_row,4); %sets Rds_on
52 Tj = Tj_Rds_on_USiC(Tj_row,1); %sets Tj
53
54 %% Matlab Model File
55 fid = fopen(component_filepath);
56 lines = textscan(fid,'%s','delimiter','\n');
57 fclose(fid);
58 lines = lines{1};
59
60 %modified lines
61 lines{18,1} = 'Ld nd nd1 5n'; %modified to maintain spacing, tab
       incidentially deleted on input
62 lines{19,1} = 'Lmd ns1 nd2 2n';
63 lines{20,1} = 'Ljg ng1 ns3 4n'; %modified to maintain spacing, tab
       incidentially deleted on input
64 lines{21,1} = 'Lmg ng ng2 20n'; %modified to maintain spacing, tab
       incidentially deleted on input
65 lines{22,1} = 'Lms ns2 ns3 3n';
66 lines{23,1} = 'Ls ns3 ns 2n';
```

```
lines{24,1} = ['xj1 nd1 ng1 ns1 jfet_G3_1200V_Ron params: Ron=' num2str(Rds_on) ' Rgoff=1.2 Rgon=1.2'];
lines{25,1} = 'xm1 nd2 ng2 ns2 mfet180';

fid = fopen(component_filepath,'w');
fprintf(fid,'%s\n',lines{:});
fclose(fid);

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value / file with values
line{02} = 'Vin             input  0                              {Vin}';
line{03} = 'XU1             input  gate   output                  UJ3C120070K3S';
line{04} = 'Rgate           gate   Vg                             {Rgate}';
line{05} = 'Vgate           Vg     output                         PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{06} = 'Rload           output 0                              {Rload}';

line{07} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart) ' ' num2str(Tmaxstep)];
line{08} = ['.lib ' component_filepath];
line{09} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e-4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{10} = '.ic V(input)={Vin} V(Vg)={VgI}';
line{11} = ['.param Vin = ' num2str(Vin_vec)];
line{12} = ['.param VgI = ' num2str(VgateI_vec)];
line{13} = ['.param VgF = ' num2str(VgateF_vec)];
line{14} = ['.param VgD = ' num2str(VgateD_vec)];
line{15} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{16} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{17} = ['.param VgO = ' num2str(VgateO_vec)];
line{18} = ['.param Rgate = ' num2str(Rgate_vec)];
line{19} = ['.step param Rload ' Rload_vec];
line{20} = '.backanno';
line{21} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "' circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. > In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
```

```
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(8,target));
            Pin = Vin.*abs(Iin);

            Vd = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs = nanmean(raw_data.variable_mat(3,target)); %same as Vout
            Vds = Vd-Vs;
            Pds = Vds.*abs(Iin);

            Efficiency = 100 * (Pin-Pds)./Pin;

            Tj = Tj;

            SWeff_uni_mat(Row,1) = abs(Iin);
            SWeff_uni_mat(Row,2) = Tj;
            SWeff_uni_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

end %ends Rds_on for loop

%% Data Save
save([data_filepath 'UJ3C120070K3S\SwitchEfficiency\SWeff_uni_mat_temp.mat'
    ],'SWeff_uni_mat');
```

Software C.7: Efficiency Simulation of a Unidirectional circuit using an Device 7

```
%% SSCB MOSFET Efficiency
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\SwitchEfficiency\SSCB_uni_Wolf
    .cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\MOSFET\Wolfspeed\']; %data file path
circuit_filepath = [data_filepath 'C2M0080120D\SwitchEfficiency\
    SSCB_uni_Wolf']; %circuit simulation file path
component_filepath = [data_filepath 'C2M0080120D_temp.lib']; %switch model
    file path

```

```
%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 2.5; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_uni_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value /
    file with values
line{02} = 'Vin             input  0                              {Vin}';
line{03} = 'XU1             input  gate   output Tj     Tc       C2M0080120D
    ';
line{05} = 'Rgate           gate   Vg                             {Rgate}';
line{06} = 'Vgate           Vg     output                         PULSE({VgI}
     {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{07} = 'Rload           output 0                              {Rload}';
line{08} = 'VTj             Tj     0                              {VTj}';
line{09} = 'VTc             Tc     0                              {VTc}';

line{10} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep)];
line{11} = ['.lib ' component_filepath];
line{12} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{13} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg)={VgI}';
line{14} = ['.param Vin = ' num2str(Vin_vec)];
line{15} = ['.param VgI = ' num2str(VgateI_vec)];
line{16} = ['.param VgF = ' num2str(VgateF_vec)];
line{17} = ['.param VgD = ' num2str(VgateD_vec)];
line{18} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{19} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{20} = ['.param VgO = ' num2str(VgateO_vec)];
line{21} = ['.param Rgate = ' num2str(Rgate_vec)];
line{22} = ['.step param Rload ' Rload_vec];
line{23} = ['.step param VTj list ' num2str(Tj_vec)];
```

```
line{24} = ['.param VTc = ' num2str(Tc_vec)];
line{25} = '.backanno';
line{26} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(12,target));
            Pin = Vin.*abs(Iin);

            Vd = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs = nanmean(raw_data.variable_mat(2,target)); %same as Vout
            Vds = Vd-Vs;
            Pds = Vds.*abs(Iin);

            Efficiency = 100 * (Pin-Pds)./Pin;

            Tj = nanmean(raw_data.variable_mat(4,target)); %per wolfspeed
    doc VTj = Tj

            SWeff_uni_mat(Row,1) = abs(Iin);
            SWeff_uni_mat(Row,2) = Tj;
            SWeff_uni_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

%% Data Save
save([data_filepath 'C2M0080120D\SwitchEfficiency\SWeff_uni_mat_temp.mat'],'
    SWeff_uni_mat');
```

Software C.8: Efficiency Simulation of a Unidirectional circuit using Device 8

```
% SSCB MOSFET Efficiency
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C3M0075120D\SwitchEfficiency\SSCB_uni_Wolf2.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C3M0075120D_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\MOSFET\Wolfspeed\']; %data file path
circuit_filepath = [data_filepath 'C3M0075120D\SwitchEfficiency\SSCB_uni_Wolf']; %circuit simulation file path
component_filepath = [data_filepath 'C3M0075120D_temp.lib']; %switch model file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -4; %initial gate voltage, -4
VgateF_vec = 15; %final gate voltage, 15
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 17 *1e-9; %rise time
VgateTf_vec = 13 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_uni_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value / file with values
line{02} = 'Vin             input  0                             {Vin}';
line{03} = 'XU1             input  Vg     output Tj     Tc       C3M0075120D';
line{04} = 'Vgate           Vg     output                        PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{05} = 'Rload           output 0                             {Rload}';
```

```
line{06} = 'VTj             Tj      0                                   {VTj}';
line{07} = 'VTc             Tc      0                                   {VTc}';

line{08} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop) ' ' num2str(Tstart)
    ' ' num2str(Tmaxstep)];
line{09} = ['.lib ' component_filepath];
line{10} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{11} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg)={VgI}';
line{12} = ['.param Vin = ' num2str(Vin_vec)];
line{13} = ['.param VgI = ' num2str(VgateI_vec)];
line{14} = ['.param VgF = ' num2str(VgateF_vec)];
line{15} = ['.param VgD = ' num2str(VgateD_vec)];
line{16} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{17} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{18} = ['.param VgO = ' num2str(VgateO_vec)];
line{19} = ['.step param Rload ' Rload_vec];
line{20} = ['.step param VTj list ' num2str(Tj_vec)];
line{21} = ['.param VTc = ' num2str(Tc_vec)];
line{22} = '.backanno';
line{23} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(10,target));
            Pin = Vin.*abs(Iin);

            Vd = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs = nanmean(raw_data.variable_mat(2,target)); %same as Vout
            Vds = Vd-Vs;
            Pds = Vds.*abs(Iin);

            Efficiency = 100 * (Pin-Pds)./Pin;

            Tj = nanmean(raw_data.variable_mat(4,target)); %per wolfspeed
    doc VTj = Tj
```

```
104
105             SWeff_uni_mat(Row,1) = abs(Iin);
106             SWeff_uni_mat(Row,2) = Tj;
107             SWeff_uni_mat(Row,3) = Efficiency;
108
109             Row = Row+1; %increments to next row for data dump
110             clear target
111             span = 1;
112         end
113     end
114 end
115
116 %% Data Save
117 save([data_filepath 'C3M0075120D\SwitchEfficiency\SWeff_uni_mat_temp.mat'],'
    SWeff_uni_mat');
```

Software C.9: Efficiency Simulation of a Unidirectional circuit using Device 9

```
1  %% SSCB MOSFET Efficiency
2  clc
3  close all
4  clear variables
5
6  %% Paths
7  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\IGBT\Infineon\IKW15N120BH6\SwitchEfficiency\SSCB_uni_Inf.
       cir
8  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\IGBT\Infineon\IGBT_1200V_TRENCHSTOP_IGBT6_L1_temp.lib
9  %C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe
10
11 %% Setup
12 parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
       Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
13 data_filepath = [parent_filepath '\IGBT\Infineon\']; %data file path
14 circuit_filepath = [data_filepath 'IKW15N120BH6\SwitchEfficiency\
       SSCB_uni_Inf']; %circuit simulation file path
15 component_filepath = [data_filepath 'IGBT_1200V_TRENCHSTOP_IGBT6_L1_temp.lib
       ']; %switch model file path
16
17 %all parameters are scriptable, use .step param or .step param XXX list
18 Tprint = 0;
19 Tstop = 1; %transient analysis stop time
20 Tstart = 0;
21 Tmaxstep = 1 *1e-3; %transient analysis step
22 AVGstart = 0.9; %start averaging
23
24 Vin_vec = 1000; %input voltage
25
26 VgateI_vec = 0; %initial gate voltage
27 VgateF_vec = 15; %final gate voltage
28 VgateD_vec = 0.5; %gate voltage delay
29 VgateTr_vec = 29 *1e-9; %rise time
30 VgateTf_vec = 63 *1e-9; %fall time
31 VgateO_vec = 0.5; %gate voltage ON time
32
33 Rgate_vec = 22; %settings from datasheet
34
```

```
Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_uni_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value /
    file with values
line{02} = 'Vin             input  0                              {Vin}';
line{03} = 'XU1             input  gate   output
    IKW15N120BH6_L1 TJ={VTj}';
line{04} = 'Rgate           gate   Vg                             {Rgate}';
line{05} = 'Vgate           Vg     output                         PULSE({VgI}
     {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet, Tr is
    middle of 7n 17n
line{06} = 'Rload           output 0                              {Rload}';

line{07} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop) ' ' num2str(Tstart)
    ' ' num2str(Tmaxstep)];
line{08} = ['.lib ' component_filepath];
line{09} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{10} = '.ic V(input)={Vin} V(Vg)={VgI}';
line{11} = ['.param Vin = ' num2str(Vin_vec)];
line{12} = ['.param VgI = ' num2str(VgateI_vec)];
line{13} = ['.param VgF = ' num2str(VgateF_vec)];
line{14} = ['.param VgD = ' num2str(VgateD_vec)];
line{15} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{16} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{17} = ['.param VgO = ' num2str(VgateO_vec)];
line{18} = ['.param Rgate = ' num2str(Rgate_vec)];
line{19} = ['.step param Rload ' Rload_vec];
line{20} = ['.step param VTj list ' num2str(Tj_vec)];
line{21} = '.backanno';
line{22} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);
```

```
85
86 %UniDirectional Circuit
87 time = raw_data.time_vect;
88 for ii = 1:size(time,2)
89     if time(ii) >= AVGstart
90         target(span) = ii;
91         span = span+1;
92         if time(ii) == Tstop
93             Vin = nanmean(raw_data.variable_mat(1,target));
94             Iin = nanmean(raw_data.variable_mat(8,target));
95             Pin = Vin.*abs(Iin);
96
97             Vc = nanmean(raw_data.variable_mat(1,target)); %same as Vin
98             Ve = nanmean(raw_data.variable_mat(2,target)); %same as Vout
99             Vce = Vc-Ve;
100            Pce = Vce.*abs(Iin);
101
102            Efficiency = 100 * (Pin-Pce)./Pin;
103
104 %             Tj = nanmean(raw_data.variable_mat(4,target)); %per doc VTj =
    Tj
105
106            SWeff_uni_mat(Row,1) = abs(Iin);
107 %             SWeff_uni_mat(Row,2) = Tj;
108            SWeff_uni_mat(Row,3) = Efficiency;
109
110            Row = Row+1; %increments to next row for data dump
111            clear target
112            span = 1;
113        end
114    end
115 end
116
117 SWeff_uni_mat(1:71,2) = Tj_vec(1);
118 SWeff_uni_mat(72:142,2) = Tj_vec(2);
119 SWeff_uni_mat(143:213,2) = Tj_vec(3);
120 SWeff_uni_mat(214:284,2) = Tj_vec(4);
121
122 %% Data Save
123 save([data_filepath 'IKW15N120BH6\SwitchEfficiency\SWeff_uni_mat_temp.mat'],
    'SWeff_uni_mat');
```

Software C.10: Efficiency Simulation of a Unidirectional circuit using Device 10

```
1 %% SSCB MOSFET Efficiency
2 clc
3 close all
4 clear variables
5
6 %% Paths
7 %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\IGBT\Infineon\IKW15N120T2\SwitchEfficiency\SSCB_uni_Inf.cir
8 %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\IGBT\Infineon\IGBT_1200V_TRENCHSTOP_IGBT6_L1_temp.lib
9 %C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe
10
11 %% Setup
```

```
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\IGBT\Infineon\']; %data file path
circuit_filepath = [data_filepath 'IKW15N120T2\SwitchEfficiency\SSCB_uni_Inf']; %circuit simulation file path
component_filepath = [data_filepath 'IGBT_1200V_TRENCHSTOP2_L1_temp.lib']; %switch model file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = 0; %initial gate voltage
VgateF_vec = 15; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 25 *1e-9; %rise time
VgateTf_vec = 95 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 41.8; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_uni_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value / file with values
line{02} = 'Vin             input  0                             {Vin}';
line{03} = 'XU1             input  gate   output                 IKW15N120T2_L1 TJ={VTj}';
line{04} = 'Rgate           gate   Vg                           {Rgate}';
line{05} = 'Vgate           Vg     output                        PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet, Tr is middle of 7n 17n
line{06} = 'Rload           output 0                             {Rload}';

line{07} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop) ' ' num2str(Tstart) ' ' num2str(Tmaxstep)];
line{08} = ['.lib ' component_filepath];
line{09} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e-4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{10} = '.ic V(input)={Vin} V(Vg)={VgI}';
line{11} = ['.param Vin = ' num2str(Vin_vec)];
line{12} = ['.param VgI = ' num2str(VgateI_vec)];
line{13} = ['.param VgF = ' num2str(VgateF_vec)];
```

```
line{14} = ['.param VgD = ' num2str(VgateD_vec)];
line{15} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{16} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{17} = ['.param VgO = ' num2str(VgateO_vec)];
line{18} = ['.param Rgate = ' num2str(Rgate_vec)];
line{19} = ['.step param Rload ' Rload_vec];
line{20} = ['.step param VTj list ' num2str(Tj_vec)];
line{21} = '.backanno';
line{22} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(8,target));
            Pin = Vin.*abs(Iin);

            Vc = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Ve = nanmean(raw_data.variable_mat(2,target)); %same as Vout
            Vce = Vc-Ve;
            Pce = Vce.*abs(Iin);

            Efficiency = 100 * (Pin-Pce)./Pin;

%             Tj = nanmean(raw_data.variable_mat(4,target)); %per doc VTj =
    Tj

            SWeff_uni_mat(Row,1) = abs(Iin);
%             SWeff_uni_mat(Row,2) = Tj;
            SWeff_uni_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

```

```
SWeff_uni_mat(1:71,2) = Tj_vec(1);
SWeff_uni_mat(72:142,2) = Tj_vec(2);
SWeff_uni_mat(143:213,2) = Tj_vec(3);
SWeff_uni_mat(214:284,2) = Tj_vec(4);

%% Data Save
save([data_filepath 'IKW15N120T2\SwitchEfficiency\SWeff_uni_mat_temp.mat'],'
    SWeff_uni_mat');
```

Software C.11: Efficiency Simulation of a Unidirectional circuit using Device 11

```
%% SSCB MOSFET Efficiency
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Infineon\IMW120R060M1H\SwitchEfficiency\SSCB_bi_Inf.
    cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Infineon\CoolSiC_MOSFET_1200V_G1_30-350mOhm_1.1_temp
    .lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\MOSFET\Infineon\']; %data file path
circuit_filepath = [data_filepath 'IMW120R060M1H\SwitchEfficiency\
    SSCB_bi_Inf']; %circuit simulation file path
component_filepath = [data_filepath 'CoolSiC_MOSFET_1200V_G1_30-350mOhm_1.1
    _temp.lib']; %switch model file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -0.1; %initial gate voltage
VgateF_vec = 18; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 17 *1e-9; %rise time
VgateTf_vec = 12 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 2; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
```

```
target = zeros(1,1); %initializes target time matrix
SWeff_bi_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                {Vin
    }';
line{03} = 'XU1           input     gate_U1   TIE       Tj     Tc
    IMW120R060M1H_L3';
line{04} = 'Rgate_U1      gate_U1   Vg1                              {
    Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{06} = 'XU2           output    gate_U2   TIE       Tj     Tc
    IMW120R060M1H_L3';
line{07} = 'Rgate_U2      gate_U2   Vg2                              {
    Rgate}';
line{08} = 'Vgate_U2      Vg2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{09} = 'Rload         output    0                                {
    Rload}';
line{10} = 'VTj           Tj        0                                {VTj
    }';
line{11} = 'VTc           Tc        0                                {VTc
    }';
line{12} = 'C1            Tj        0                                1pf'
    ;
line{13} = 'C2            Tc        0                                1pf'
    ;

line{14} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop) ' ' num2str(Tstart)
    ' ' num2str(Tmaxstep)];
line{15} = ['.lib ' component_filepath];
line{16} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL4=500 SRCSTEPS=100';
line{17} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg1)={VgI} V(Vg2)={
    VgI}';
line{18} = ['.param Vin = ' num2str(Vin_vec)];
line{19} = ['.param VgI = ' num2str(VgateI_vec)];
line{20} = ['.param VgF = ' num2str(VgateF_vec)];
line{21} = ['.param VgD = ' num2str(VgateD_vec)];
line{22} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{23} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{24} = ['.param VgO = ' num2str(VgateO_vec)];
line{25} = ['.param Rgate = ' num2str(Rgate_vec)];
line{26} = ['.step param Rload ' Rload_vec];
line{27} = ['.step param VTj list ' num2str(Tj_vec)];
line{28} = ['.param VTc = ' num2str(Tc_vec)];
line{29} = '.backanno';
line{30} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
```

```
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(19,target));
            Pin = Vin.*abs(Iin);

            Vd_U1 = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs_U1 = nanmean(raw_data.variable_mat(3,target)); %same as Vtie
            Vds_U1 = Vd_U1-Vs_U1;
            Pds_U1 = Vds_U1.*abs(Iin);

            Vd_U2 = nanmean(raw_data.variable_mat(8,target)); %same as Vout
            Vs_U2 = nanmean(raw_data.variable_mat(3,target)); %same as Vtie
            Vsd_U2 = Vs_U2-Vd_U2;
            Psd_U2 = Vsd_U2.*abs(Iin);

            Efficiency = 100 * (Pin-(Pds_U1+Psd_U2))./Pin;

            Tj = nanmean(raw_data.variable_mat(4,target)); %per doc VTj = Tj

            SWeff_bi_mat(Row,1) = abs(Iin);
            SWeff_bi_mat(Row,2) = Tj;
            SWeff_bi_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end


%% Data Save
save([data_filepath 'IMW120R060M1H\SwitchEfficiency\SWeff_bi_mat_temp.mat'],
    'SWeff_bi_mat');
```

Software C.12: Efficiency Simulation of a Bidirectional circuit using an Device 1

```
%% SSCB MOSFET Efficiency
clc
```

```
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Littelfuse\LSIC1MO120E0080\SwitchEfficiency\SSCB_bi_Littelfuse.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Littelfuse\LSIC1MO120E0080\LSIC1MO120E0080_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\MOSFET\Littelfuse\']; %data file path
circuit_filepath = [data_filepath 'LSIC1MO120E0080\SwitchEfficiency\SSCB_bi_Littelfuse']; %circuit simulation file path
component_filepath = [data_filepath 'LSIC1MO120E0080_temp.lib']; %switch model file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 10 *1e-9; %rise time
VgateTf_vec = 6 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 1.9; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_bi_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5          value / file with values
line{02} = 'Vin           input     0                                        {Vin}';
line{03} = 'XU1           input     gate_U1   TIE                            LSIC1MO120E0080 T = {Tj}';
line{04} = 'Rgate_U1      gate_U1   Vg1                                      {Rgate}';
```

```
line{05} = 'Vgate_U1       Vg1        TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{06} = 'XU2            output     gate_U2    TIE
    LSIC1MO120E0080 T = {Tj}';
line{07} = 'Rgate_U2       gate_U2    Vg2                                    {
    Rgate}';
line{08} = 'Vgate_U2       Vg2        TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{09} = 'Rload          output     0                                      {
    Rload}';
line{10} = 'VTj            Tj         0                                      {Tj}
    ';

line{11} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep)];
line{12} = ['.lib ' component_filepath];
line{13} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{14} = '.ic V(input)={Vin} V(Vg1)={VgI} V(Vg2)={VgI}';
line{15} = ['.param Vin = ' num2str(Vin_vec)];
line{16} = ['.param VgI = ' num2str(VgateI_vec)];
line{17} = ['.param VgF = ' num2str(VgateF_vec)];
line{18} = ['.param VgD = ' num2str(VgateD_vec)];
line{19} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{20} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{21} = ['.param VgO = ' num2str(VgateO_vec)];
line{22} = ['.param Rgate = ' num2str(Rgate_vec)];
line{23} = ['.step param Rload ' Rload_vec];
line{24} = ['.step param Tj list ' num2str(Tj_vec)];
line{25} = '.backanno';
line{26} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(15,target));
            Pin = Vin.*abs(Iin);
```

```
100
101             Vd_U1 = nanmean(raw_data.variable_mat(1,target)); %same as Vin
102             Vs_U1 = nanmean(raw_data.variable_mat(2,target)); %same as Vtie
103             Vds_U1 = Vd_U1-Vs_U1;
104             Pds_U1 = Vds_U1.*abs(Iin);
105
106             Vd_U2 = nanmean(raw_data.variable_mat(5,target)); %same as Vout
107             Vs_U2 = nanmean(raw_data.variable_mat(2,target)); %same as Vtie
108             Vsd_U2 = Vs_U2-Vd_U2;
109             Psd_U2 = Vsd_U2.*abs(Iin);
110
111             Efficiency = 100 * (Pin-(Pds_U1+Psd_U2))./Pin;
112
113             Tj = nanmean(raw_data.variable_mat(8,target));
114
115             SWeff_bi_mat(Row,1) = abs(Iin);
116             SWeff_bi_mat(Row,2) = Tj;
117             SWeff_bi_mat(Row,3) = Efficiency;
118
119             Row = Row+1; %increments to next row for data dump
120             clear target
121             span = 1;
122         end
123     end
124 end
125
126 %% Data Save
127 save([data_filepath 'LSIC1MO120E0080\SwitchEfficiency\SWeff_bi_mat_temp.mat'
    ],'SWeff_bi_mat');
```

Software C.13: Efficiency Simulation of a Bidirectional circuit using Device 2

```
1  %% SSCB MOSFET Efficiency
2  clc
3  close all
4  clear variables
5
6  %% Paths
7  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\MOSFET\MicrochipTech\MSC080SMA120J\SwitchEfficiency\
       SSCB_bi_MicroT.cir
8  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\MOSFET\MicrochipTech\MSC080SMA120J.lib
9  %C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe
10
11 %% Setup
12 parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
       Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
13 data_filepath = [parent_filepath '\MOSFET\MicrochipTech\']; %data file path
14 circuit_filepath = [data_filepath 'MSC080SMA120J\SwitchEfficiency\
       SSCB_bi_MicroT']; %circuit simulation file path
15 component_filepath = [data_filepath 'MSCxxxSMA12020190212_temp.lib']; %
       switch model file path
16
17 %all parameters are scriptable, use .step param or .step param XXX list
18 Tprint = 0;
19 Tstop = 1; %transient analysis stop time
20 Tstart = 0;
```

```
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 4 *1e-9; %rise time
VgateTf_vec = 15 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 4; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 100 *1e6; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_bi_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                {Vin
    }';
line{03} = 'XU1           input     gate_U1   TIE       Tj     Tc
    MSC080SMA120Jt';
line{04} = 'Rgate_U1      gate_U1   Vg1                              {
    Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{06} = 'XU2           output    gate_U2   TIE       Tj     Tc
    MSC080SMA120Jt';
line{07} = 'Rgate_U2      gate_U2   Vg2                              {
    Rgate}';
line{08} = 'Vgate_U2      Vg2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{09} = 'Rload         output    0                                {
    Rload}';
line{10} = 'VTj           Tj        0                                {VTj
    }';
line{11} = 'RTc           Tc        0                                {RTc
    }';

line{12} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep)];
line{13} = ['.lib ' component_filepath];
line{14} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{15} = '.ic V(input)={Vin} V(Tj)={VTj} V(Vg1)={VgI} V(Vg2)={VgI}';
line{16} = ['.param Vin = ' num2str(Vin_vec)];
line{17} = ['.param VgI = ' num2str(VgateI_vec)];
line{18} = ['.param VgF = ' num2str(VgateF_vec)];
```

```
line{19} = ['.param VgD = ' num2str(VgateD_vec)];
line{20} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{21} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{22} = ['.param VgO = ' num2str(VgateO_vec)];
line{23} = ['.param Rgate = ' num2str(Rgate_vec)];
line{24} = ['.step param Rload ' Rload_vec];
line{25} = ['.step param VTj list ' num2str(Tj_vec)];
line{26} = ['.param RTc = ' num2str(Tc_vec)];
line{27} = '.backanno';
line{28} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(17,target));
            Pin = Vin.*abs(Iin);

            Vd_U1 = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs_U1 = nanmean(raw_data.variable_mat(3,target)); %same as Vtie
            Vds_U1 = Vd_U1-Vs_U1;
            Pds_U1 = Vds_U1.*abs(Iin);

            Vd_U2 = nanmean(raw_data.variable_mat(8,target)); %same as Vout
            Vs_U2 = nanmean(raw_data.variable_mat(3,target)); %same as Vtie
            Vsd_U2 = Vs_U2-Vd_U2;
            Psd_U2 = Vsd_U2.*abs(Iin);

            Efficiency = 100 * (Pin-(Pds_U1+Psd_U2))./Pin;

            Tj = nanmean(raw_data.variable_mat(4,target)); %per doc VTj = Tj

            SWeff_bi_mat(Row,1) = abs(Iin);
            SWeff_bi_mat(Row,2) = Tj;
            SWeff_bi_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
```

```
            span = 1;
        end
    end
end

%% Data Save
save([data_filepath 'MSC080SMA120J\SwitchEfficiency\SWeff_bi_mat_temp.mat'],
    'SWeff_bi_mat');
```

Software C.14: Efficiency Simulation of a Bidirectional circuit using Device 3

```
%% SSCB MOSFET Efficiency
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Rohm\SCT3080KLGC11\SwitchEfficiency\SSCB_bi_Rohm.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Rohm\SCT3080KL_LT_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\MOSFET\Rohm\']; %data file path
circuit_filepath = [data_filepath 'SCT3080KLGC11\SwitchEfficiency\
    SSCB_bi_Rohm']; %circuit simulation file path
component_filepath = [data_filepath 'SCT3080KL_LT_temp.lib']; %switch model
    file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = 0; %initial gate voltage
VgateF_vec = 18; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 15 *1e-9; %rise time
VgateTf_vec = 24 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_bi_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

```

```
%% Temperature Modification
for iii = 1:size(Tj_vec,2)

T0 = Tj_vec(iii); %sets Tj

%% Matlab Model File
fid = fopen(component_filepath);
lines = textscan(fid,'%s','delimiter','\n');
fclose(fid);
lines = lines{1};

%modified lines
lines{10,1} = ['.PARAM T0=' num2str(T0)];

fid = fopen(component_filepath,'w');
fprintf(fid,'%s\n',lines{:});
fclose(fid);

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                {Vin
    }';
line{03} = 'XU1           input     Vg1       TIE
    SCT3080KL_LT';
line{04} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{05} = 'XU2           output    Vg2       TIE
    SCT3080KL_LT';
line{06} = 'Vgate_U2      Vg2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{07} = 'Rload         output    0                                {
    Rload}';

line{08} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep)];
line{09} = ['.lib ' component_filepath];
line{10} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{11} = '.ic V(input)={Vin} V(Vg1)={VgI} V(Vg2)={VgI}';
line{12} = ['.param Vin = ' num2str(Vin_vec)];
line{13} = ['.param VgI = ' num2str(VgateI_vec)];
line{14} = ['.param VgF = ' num2str(VgateF_vec)];
line{15} = ['.param VgD = ' num2str(VgateD_vec)];
line{16} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{17} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{18} = ['.param VgO = ' num2str(VgateO_vec)];
line{19} = ['.step param Rload ' Rload_vec];
line{20} = '.backanno';
line{21} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);
```

```
93
94 %% Simulate
95 system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
       circuit_filepath '.cir"']);
96
97 %% Data Collection
98 warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
       encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
        In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
99 raw_data = LTspice2Matlab([circuit_filepath '.raw']);
100
101 %BiDirectional Circuit
102 time = raw_data.time_vect;
103 for ii = 1:size(time,2)
104     if time(ii) >= AVGstart
105         target(span) = ii;
106         span = span+1;
107         if time(ii) == Tstop
108             Vin = nanmean(raw_data.variable_mat(1,target));
109             Iin = nanmean(raw_data.variable_mat(9,target));
110             Pin = Vin.*abs(Iin);
111
112             Vd_U1 = nanmean(raw_data.variable_mat(1,target)); %same as Vin
113             Vs_U1 = nanmean(raw_data.variable_mat(3,target)); %same as Vtie
114             Vds_U1 = Vd_U1-Vs_U1;
115             Pds_U1 = Vds_U1.*abs(Iin);
116
117             Vd_U2 = nanmean(raw_data.variable_mat(4,target)); %same as Vout
118             Vs_U2 = nanmean(raw_data.variable_mat(3,target)); %same as Vtie
119             Vsd_U2 = Vs_U2-Vd_U2;
120             Psd_U2 = Vsd_U2.*abs(Iin);
121
122             Efficiency = 100 * (Pin-(Pds_U1+Psd_U2))./Pin;
123
124             Tj = T0;
125
126             SWeff_bi_mat(Row,1) = abs(Iin);
127             SWeff_bi_mat(Row,2) = Tj;
128             SWeff_bi_mat(Row,3) = Efficiency;
129
130             Row = Row+1; %increments to next row for data dump
131             clear target
132             span = 1;
133         end
134     end
135 end
136
137 end %ends Rds_on for loop
138
139 %% Data Save
140 save([data_filepath 'SCT3080KLGC11\SwitchEfficiency\SWeff_bi_mat_temp.mat'],
       'SWeff_bi_mat');
```

Software C.15: Efficiency Simulation of a Bidirectional circuit using Device 4

```
1 %% SSCB JFET Efficiency
2 clc
3 close all
```

```
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\JFET\UnitedSiC\UJ3N120070K3S\SwitchEfficiency\SSCB_bi_USiC.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\JFET\UnitedSiC\UJ3N120070K3S_temp.301
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\JFET\UnitedSiC\']; %data file path
circuit_filepath = [data_filepath 'UJ3N120070K3S\SwitchEfficiency\SSCB_bi_USiC']; %circuit simulation file path
component_filepath = [data_filepath 'UJ3N120070K3S_temp.301']; %switch model file path

load([data_filepath 'Tj_Rds_on_USiC.mat']); %temp curve for switch

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-4; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -18; %initial gate voltage
VgateF_vec = 0; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 25 *1e-9; %rise time
VgateTf_vec = 39 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 1; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [1 81 206 231]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_bi_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% Rds_on Modification
for iii = 1:size(Tj_vec,2)

Tj_row = Tj_vec(iii); %sets Tj
Rds_on = Tj_Rds_on_USiC(Tj_row,3); %sets Rds_on
Tj = Tj_Rds_on_USiC(Tj_row,1); %sets Tj

%% Matlab Model File
fid = fopen(component_filepath);
lines = textscan(fid,'%s','delimiter','\n');
```

```
fclose(fid);
lines = lines{1};

%modified lines
lines{19,1} = 'Ld nd nd1 5n'; %modified to maintain spacing, tab
    incidentially deleted on input
lines{20,1} = 'Ljg ng ng1 10n'; %modified to maintain spacing, tab
    incidentially deleted on input
lines{21,1} = 'Ls ns ns1 2n'; %modified to maintain spacing, tab
    incidentially deleted on input
lines{22,1} = ['xj1 nd1 ng1 ns1 jfet_G3_1200V_Ron params: Ron=' num2str(
    Rds_on) ' Rgoff=3.5 Rgon=3.5'];

fid = fopen(component_filepath,'w');
fprintf(fid,'%s\n',lines{:});
fclose(fid);

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                {Vin
    }';
line{03} = 'XU1           input     gate_U1   TIE
    UJ3N120070K3S';
line{04} = 'Rgate_U1      gate_U1   Vg1                              {
    Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{06} = 'XU2           output    gate_U2   TIE
    UJ3N120070K3S';
line{07} = 'Rgate_U2      gate_U2   Vg2                              {
    Rgate}';
line{08} = 'Vgate_U2      Vg2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{09} = 'Rload         output    0                                {
    Rload}';

line{10} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep)];
line{11} = ['.lib ' component_filepath];
line{12} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{13} = '.ic V(input)={Vin} V(Vg1)={VgI} V(Vg2)={VgI}';
line{14} = ['.param Vin = ' num2str(Vin_vec)];
line{15} = ['.param VgI = ' num2str(VgateI_vec)];
line{16} = ['.param VgF = ' num2str(VgateF_vec)];
line{17} = ['.param VgD = ' num2str(VgateD_vec)];
line{18} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{19} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{20} = ['.param VgO = ' num2str(VgateO_vec)];
line{21} = ['.param Rgate = ' num2str(Rgate_vec)];
line{22} = ['.step param Rload ' Rload_vec];
line{23} = '.backanno';
line{24} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
```

```
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(13,target));
            Pin = Vin.*abs(Iin);

            Vd_U1 = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs_U1 = nanmean(raw_data.variable_mat(3,target)); %same as Vtie
            Vds_U1 = Vd_U1-Vs_U1;
            Pds_U1 = Vds_U1.*abs(Iin);

            Vd_U2 = nanmean(raw_data.variable_mat(5,target)); %same as Vout
            Vs_U2 = nanmean(raw_data.variable_mat(3,target)); %same as Vtie
            Vsd_U2 = Vs_U2-Vd_U2;
            Psd_U2 = Vsd_U2.*abs(Iin);

            Efficiency = 100 * (Pin-(Pds_U1+Psd_U2))./Pin;

            Tj = Tj;

            SWeff_bi_mat(Row,1) = abs(Iin);
            SWeff_bi_mat(Row,2) = Tj;
            SWeff_bi_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

end %ends Rds_on for loop

%% Data Save
save([data_filepath 'UJ3N120070K3S\SwitchEfficiency\SWeff_bi_mat_temp.mat'],
    'SWeff_bi_mat');
```

Software C.16: Efficiency Simulation of a Bidirectional circuit using an Device 5

```
%% SSCB JFET Efficiency
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\JFET\UnitedSiC\UJ3C120080K3S\SwitchEfficiency\SSCB_bi_USiC.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\JFET\UnitedSiC\UJ3C120080K3S_temp.301
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\JFET\UnitedSiC\']; %data file path
circuit_filepath = [data_filepath 'UJ3C120080K3S\SwitchEfficiency\SSCB_bi_USiC']; %circuit simulation file path
component_filepath = [data_filepath 'UJ3C120080K3S_temp.301']; %switch model file path

load([data_filepath 'Tj_Rds_on_USiC.mat']); %temp curve for switch

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-4; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 15; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 14 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 1; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [1 81 206 231]; %sets Tj row vector, corresponds to -55, 25, 150, 175
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_bi_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% Rds_on Modification
for iii = 1:size(Tj_vec,2)

Tj_row = Tj_vec(iii); %sets Tj
Rds_on = Tj_Rds_on_USiC(Tj_row,3); %sets Rds_on
Tj = Tj_Rds_on_USiC(Tj_row,1); %sets Tj
```

```
53
54 %% Matlab Model File
55 fid = fopen(component_filepath);
56 lines = textscan(fid,'%s','delimiter','\n');
57 fclose(fid);
58 lines = lines{1};
59
60 %modified lines
61 lines{18,1} = 'Ld nd nd1 5n'; %modified to maintain spacing, tab
       incidentially deleted on input
62 lines{19,1} = 'Lmd ns1 nd2 2n';
63 lines{20,1} = 'Ljg ng1 ns3 4n'; %modified to maintain spacing, tab
       incidentially deleted on input
64 lines{21,1} = 'Lmg ng ng2 70n'; %modified to maintain spacing, tab
       incidentially deleted on input
65 lines{22,1} = 'Lms ns2 ns3 3n';
66 lines{23,1} = 'Ls ns3 ns 2n';
67 lines{24,1} = ['xj1 nd1 ng1 ns1 jfet_G3_1200V_Ron params: Ron=' num2str(
       Rds_on) ' Rgoff=1.7 Rgon=1.7'];
68 lines{25,1} = 'xm1 nd2 ng2 ns2 mfet180';
69
70 fid = fopen(component_filepath,'w');
71 fprintf(fid,'%s\n',lines{:});
72 fclose(fid);
73
74 %% MatLab Netlist
75 line{01} = ['* ' circuit_filepath '.asc'];
76 %BiDirectional Circuit
77 %           name          node1     node2     node3     node4  node5
       value / file with values
78 line{02} = 'Vin           input     0                                  {Vin
       }';
79 line{03} = 'XU1           input     gate_U1   TIE
       UJ3C120080K3S';
80 line{04} = 'Rgate_U1      gate_U1   Vg1                                {
       Rgate}';
81 line{05} = 'Vgate_U1      Vg1       TIE
       PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
82 line{06} = 'XU2           output    gate_U2   TIE
       UJ3C120080K3S';
83 line{07} = 'Rgate_U2      gate_U2   Vg2                                {
       Rgate}';
84 line{08} = 'Vgate_U2      Vg2       TIE
       PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
85 line{09} = 'Rload         output    0                                  {
       Rload}';
86
87 line{10} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
       ' ' num2str(Tmaxstep)];
88 line{11} = ['.lib ' component_filepath];
89 line{12} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
       -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
90 line{13} = '.ic V(input)={Vin} V(Vg1)={VgI} V(Vg2)={VgI}';
91 line{14} = ['.param Vin = ' num2str(Vin_vec)];
92 line{15} = ['.param VgI = ' num2str(VgateI_vec)];
93 line{16} = ['.param VgF = ' num2str(VgateF_vec)];
94 line{17} = ['.param VgD = ' num2str(VgateD_vec)];
95 line{18} = ['.param VgTr = ' num2str(VgateTr_vec)];
96 line{19} = ['.param VgTf = ' num2str(VgateTf_vec)];
```

```
line{20} = ['.param VgO = ' num2str(VgateO_vec)];
line{21} = ['.param Rgate = ' num2str(Rgate_vec)];
line{22} = ['.step param Rload ' Rload_vec];
line{23} = '.backanno';
line{24} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(13,target));
            Pin = Vin.*abs(Iin);

            Vd_U1 = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs_U1 = nanmean(raw_data.variable_mat(3,target)); %same as Vtie
            Vds_U1 = Vd_U1-Vs_U1;
            Pds_U1 = Vds_U1.*abs(Iin);

            Vd_U2 = nanmean(raw_data.variable_mat(5,target)); %same as Vout
            Vs_U2 = nanmean(raw_data.variable_mat(3,target)); %same as Vtie
            Vsd_U2 = Vs_U2-Vd_U2;
            Psd_U2 = Vsd_U2.*abs(Iin);

            Efficiency = 100 * (Pin-(Pds_U1+Psd_U2))./Pin;

            Tj = Tj;

            SWeff_bi_mat(Row,1) = abs(Iin);
            SWeff_bi_mat(Row,2) = Tj;
            SWeff_bi_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

```

```
153 end %ends Rds_on for loop
154
155 %% Data Save
156 save([data_filepath 'UJ3C120080K3S\SwitchEfficiency\SWeff_bi_mat_temp.mat'],
    'SWeff_bi_mat');
```

Software C.17: Efficiency Simulation of a Bidirectional circuit using an Device 6

```
1  %% SSCB JFET Efficiency
2  clc
3  close all
4  clear variables
5
6  %% Paths
7  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\JFET\UnitedSiC\UJ3C120070K3S\SwitchEfficiency\SSCB_bi_USiC.
       cir
8  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\JFET\UnitedSiC\UJ3C120070K3S_temp.301
9  %C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe
10
11 %% Setup
12 parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
       Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
13 data_filepath = [parent_filepath '\JFET\UnitedSiC\']; %data file path
14 circuit_filepath = [data_filepath 'UJ3C120070K3S\SwitchEfficiency\
       SSCB_bi_USiC']; %circuit simulation file path
15 component_filepath = [data_filepath 'UJ3C120070K3S_temp.301']; %switch model
        file path
16
17 load([data_filepath 'Tj_Rds_on_USiC.mat']); %temp curve for switch
18
19 %all parameters are scriptable, use .step param or .step param XXX list
20 Tprint = 0;
21 Tstop = 1; %transient analysis stop time
22 Tstart = 0;
23 Tmaxstep = 1 *1e-3; %transient analysis step
24 AVGstart = 0.9; %start averaging
25
26 Vin_vec = 1000; %input voltage
27
28 VgateI_vec = -5; %initial gate voltage
29 VgateF_vec = 15; %final gate voltage
30 VgateD_vec = 0.5; %gate voltage delay
31 VgateTr_vec = 17 *1e-9; %rise time
32 VgateTf_vec = 9 *1e-9; %fall time
33 VgateO_vec = 0.5; %gate voltage ON time
34
35 Rgate_vec = 1; %settings from datasheet
36
37 Rload_vec = '30 100 1'; %sets load resistance row vector
38
39 Tj_vec = [1 81 206 231]; %sets Tj row vector
40 Tc_vec = 25; %sets Tc row vector
41
42 span = 1; %initialize average start
43 target = zeros(1,1); %initializes target time matrix
44 SWeff_bi_mat = zeros(284,3); %initializes switch efficiency matrix
```

```
Row = 1; %intitialize row count for data dump

%% Rds_on Modification
for iii = 1:size(Tj_vec,2)

Tj_row = Tj_vec(iii); %sets Tj
Rds_on = Tj_Rds_on_USiC(Tj_row,4); %sets Rds_on
Tj = Tj_Rds_on_USiC(Tj_row,1); %sets Tj

%% Matlab Model File
fid = fopen(component_filepath);
lines = textscan(fid,'%s','delimiter','\n');
fclose(fid);
lines = lines{1};

%modified lines
lines{18,1} = 'Ld nd nd1 5n'; %modified to maintain spacing, tab
    incidentially deleted on input
lines{19,1} = 'Lmd ns1 nd2 2n';
lines{20,1} = 'Ljg ng1 ns3 4n'; %modified to maintain spacing, tab
    incidentially deleted on input
lines{21,1} = 'Lmg ng ng2 20n'; %modified to maintain spacing, tab
    incidentially deleted on input
lines{22,1} = 'Lms ns2 ns3 3n';
lines{23,1} = 'Ls ns3 ns 2n';
lines{24,1} = ['xj1 nd1 ng1 ns1 jfet_G3_1200V_Ron params: Ron=' num2str(
    Rds_on) ' Rgoff=1.2 Rgon=1.2'];
lines{25,1} = 'xm1 nd2 ng2 ns2 mfet180';

fid = fopen(component_filepath,'w');
fprintf(fid,'%s\n',lines{:});
fclose(fid);

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                  {Vin
    }';
line{03} = 'XU1           input     gate_U1   TIE
    UJ3C120070K3S';
line{04} = 'Rgate_U1      gate_U1   Vg1                                {
    Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{06} = 'XU2           output    gate_U2   TIE
    UJ3C120070K3S';
line{07} = 'Rgate_U2      gate_U2   Vg2                                {
    Rgate}';
line{08} = 'Vgate_U2      Vg2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{09} = 'Rload         output    0                                  {
    Rload}';

line{10} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep)];
line{11} = ['.lib ' component_filepath];
```

```
line{12} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{13} = '.ic V(input)={Vin} V(Vg1)={VgI} V(Vg2)={VgI}';
line{14} = ['.param Vin = ' num2str(Vin_vec)];
line{15} = ['.param VgI = ' num2str(VgateI_vec)];
line{16} = ['.param VgF = ' num2str(VgateF_vec)];
line{17} = ['.param VgD = ' num2str(VgateD_vec)];
line{18} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{19} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{20} = ['.param VgO = ' num2str(VgateO_vec)];
line{21} = ['.param Rgate = ' num2str(Rgate_vec)];
line{22} = ['.step param Rload ' Rload_vec];
line{23} = '.backanno';
line{24} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(13,target));
            Pin = Vin.*abs(Iin);

            Vd_U1 = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs_U1 = nanmean(raw_data.variable_mat(3,target)); %same as Vtie
            Vds_U1 = Vd_U1-Vs_U1;
            Pds_U1 = Vds_U1.*abs(Iin);

            Vd_U2 = nanmean(raw_data.variable_mat(5,target)); %same as Vout
            Vs_U2 = nanmean(raw_data.variable_mat(3,target)); %same as Vtie
            Vsd_U2 = Vs_U2-Vd_U2;
            Psd_U2 = Vsd_U2.*abs(Iin);

            Efficiency = 100 * (Pin-(Pds_U1+Psd_U2))./Pin;

            Tj = Tj;

            SWeff_bi_mat(Row,1) = abs(Iin);
            SWeff_bi_mat(Row,2) = Tj;
```

```
144             SWeff_bi_mat(Row,3) = Efficiency;
145 
146             Row = Row+1; %increments to next row for data dump
147             clear target
148             span = 1;
149         end
150     end
151 end
152 
153 end %ends Rds_on for loop
154 
155 %% Data Save
156 save([data_filepath 'UJ3C120070K3S\SwitchEfficiency\SWeff_bi_mat_temp.mat'],
    'SWeff_bi_mat');
```

Software C.18: Efficiency Simulation of a Bidirectional circuit using an Device 7

```
1  %% SSCB MOSFET Efficiency
2  clc
3  close all
4  clear variables
5  
6  %% Paths
7  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\SwitchEfficiency\SSCB_bi_Wolf.
       cir
8  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
9  %C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe
10 
11 %% Setup
12 parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
       Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
13 data_filepath = [parent_filepath '\MOSFET\Wolfspeed\']; %data file path
14 circuit_filepath = [data_filepath 'C2M0080120D\SwitchEfficiency\SSCB_bi_Wolf
       ']; %circuit simulation file path
15 component_filepath = [data_filepath 'C2M0080120D_temp.lib']; %switch model
       file path
16 
17 %all parameters are scriptable, use .step param or .step param XXX list
18 Tprint = 0;
19 Tstop = 1; %transient analysis stop time
20 Tstart = 0;
21 Tmaxstep = 1 *1e-3; %transient analysis step
22 AVGstart = 0.9; %start averaging
23 
24 Vin_vec = 1000; %input voltage
25 
26 VgateI_vec = -5; %initial gate voltage
27 VgateF_vec = 20; %final gate voltage
28 VgateD_vec = 0.5; %gate voltage delay
29 VgateTr_vec = 22 *1e-9; %rise time
30 VgateTf_vec = 14 *1e-9; %fall time
31 VgateO_vec = 0.5; %gate voltage ON time
32 
33 Rgate_vec = 2.5; %settings from datasheet
34 
35 Rload_vec = '30 100 1'; %sets load resistance row vector
```

```
36
37 Tj_vec = [-55 25 150 175]; %sets Tj row vector
38 Tc_vec = 25; %sets Tc row vector
39
40 span = 1; %initialize average start
41 target = zeros(1,1); %initializes target time matrix
42 SWeff_bi_mat = zeros(284,3); %initializes switch efficiency matrix
43 Row = 1; %intitialize row count for data dump
44
45 %% MatLab Netlist
46 line{01} = ['* ' circuit_filepath '.asc'];
47 %BiDirectional Circuit
48 %           name          node1     node2     node3     node4  node5
    value / file with values
49 line{02} = 'Vin           input     0                                    {Vin
    }';
50 line{03} = 'XU1           input     gate_U1   TIE       Tj     Tc
    C2M0080120D';
51 line{04} = 'Rgate_U1      gate_U1   Vg1                                  {
    Rgate}';
52 line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
53 line{06} = 'XU2           output    gate_U2   TIE       Tj     Tc
    C2M0080120D';
54 line{07} = 'Rgate_U2      gate_U2   Vg2                                  {
    Rgate}';
55 line{08} = 'Vgate_U2      Vg2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
56 line{09} = 'Rload         output    0                                    {
    Rload}';
57 line{10} = 'VTj           Tj        0                                    {VTj
    }';
58 line{11} = 'VTc           Tc        0                                    {VTc
    }';
59
60 line{12} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep)];
61 line{13} = ['.lib ' component_filepath];
62 line{14} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
63 line{15} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg1)={VgI} V(Vg2)={
    VgI}';
64 line{16} = ['.param Vin = ' num2str(Vin_vec)];
65 line{17} = ['.param VgI = ' num2str(VgateI_vec)];
66 line{18} = ['.param VgF = ' num2str(VgateF_vec)];
67 line{19} = ['.param VgD = ' num2str(VgateD_vec)];
68 line{20} = ['.param VgTr = ' num2str(VgateTr_vec)];
69 line{21} = ['.param VgTf = ' num2str(VgateTf_vec)];
70 line{22} = ['.param VgO = ' num2str(VgateO_vec)];
71 line{23} = ['.param Rgate = ' num2str(Rgate_vec)];
72 line{24} = ['.step param Rload ' Rload_vec];
73 line{25} = ['.step param VTj list ' num2str(Tj_vec)];
74 line{26} = ['.param VTc = ' num2str(Tc_vec)];
75 line{27} = '.backanno';
76 line{28} = '.end';
77
78 %% New Netlist
79 fid = fopen([circuit_filepath '.cir'],'wb');
80 for i = 1:length(line)
```

```
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(17,target));
            Pin = Vin.*abs(Iin);

            Vd_U1 = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs_U1 = nanmean(raw_data.variable_mat(2,target)); %same as Vtie
            Vds_U1 = Vd_U1-Vs_U1;
            Pds_U1 = Vds_U1.*abs(Iin);

            Vd_U2 = nanmean(raw_data.variable_mat(8,target)); %same as Vout
            Vs_U2 = nanmean(raw_data.variable_mat(2,target)); %same as Vtie
            Vsd_U2 = Vs_U2-Vd_U2;
            Psd_U2 = Vsd_U2.*abs(Iin);

            Efficiency = 100 * (Pin-(Pds_U1+Psd_U2))./Pin;

            Tj = nanmean(raw_data.variable_mat(4,target)); %per doc VTj = Tj

            SWeff_bi_mat(Row,1) = abs(Iin);
            SWeff_bi_mat(Row,2) = Tj;
            SWeff_bi_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

%% Data Save
save([data_filepath 'C2M0080120D\SwitchEfficiency\SWeff_bi_mat_temp.mat'],'
    SWeff_bi_mat');
```

Software C.19: Efficiency Simulation of a Bidirectional circuit using Device 8

```
%% SSCB MOSFET Efficiency
clc
```

```
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C3M0075120D\SwitchEfficiency\SSCB_bi_Wolf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C3M0075120D_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\MOSFET\Wolfspeed\']; %data file path
circuit_filepath = [data_filepath 'C3M0075120D\SwitchEfficiency\SSCB_bi_Wolf']; %circuit simulation file path
component_filepath = [data_filepath 'C3M0075120D_temp.lib']; %switch model file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -4; %initial gate voltage, -4
VgateF_vec = 15; %final gate voltage, 15
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 17 *1e-9; %rise time
VgateTf_vec = 13 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_bi_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5          value / file with values
line{02} = 'Vin           input     0                                        {Vin}';
line{03} = 'XU1           input     Vg1       TIE       Tj     Tc          C3M0075120D';
line{04} = 'Vgate_U1      Vg1       TIE       PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet, static max -4/15, dynamic max -8/19
```

```
line{05} = 'XU2            output    Vg2       TIE       Tj     Tc
    C3M0075120D';
line{06} = 'Vgate_U2       Vg2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet,
     static max -4/15, dynamic max -8/19
line{07} = 'Rload          output    0                                    {
    Rload}';
line{08} = 'VTj            Tj        0                                    {VTj
    }';
line{09} = 'VTc            Tc        0                                    {VTc
    }';

line{10} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep)];
line{11} = ['.lib ' component_filepath];
line{12} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{13} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg1)={VgI} V(Vg2)={
    VgI}';
line{14} = ['.param Vin = ' num2str(Vin_vec)];
line{15} = ['.param VgI = ' num2str(VgateI_vec)];
line{16} = ['.param VgF = ' num2str(VgateF_vec)];
line{17} = ['.param VgD = ' num2str(VgateD_vec)];
line{18} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{19} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{20} = ['.param VgO = ' num2str(VgateO_vec)];
line{21} = ['.step param Rload ' Rload_vec];
line{22} = ['.step param VTj list ' num2str(Tj_vec)];
line{23} = ['.param VTc = ' num2str(Tc_vec)];
line{24} = '.backanno';
line{25} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(13,target));
            Pin = Vin.*abs(Iin);
```

```

            Vd_U1 = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Vs_U1 = nanmean(raw_data.variable_mat(2,target)); %same as Vtie
            Vds_U1 = Vd_U1-Vs_U1;
            Pds_U1 = Vds_U1.*abs(Iin);

            Vd_U2 = nanmean(raw_data.variable_mat(7,target)); %same as Vout
            Vs_U2 = nanmean(raw_data.variable_mat(2,target)); %same as Vtie
            Vsd_U2 = Vs_U2-Vd_U2;
            Psd_U2 = Vsd_U2.*abs(Iin);

            Efficiency = 100 * (Pin-(Pds_U1+Psd_U2))./Pin;

            Tj = nanmean(raw_data.variable_mat(4,target)); %per doc VTj = Tj

            SWeff_bi_mat(Row,1) = abs(Iin);
            SWeff_bi_mat(Row,2) = Tj;
            SWeff_bi_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

%% Data Save
save([data_filepath 'C3M0075120D\SwitchEfficiency\SWeff_bi_mat_temp.mat'],'
    SWeff_bi_mat');
```

Software C.20: Efficiency Simulation of a Bidirectional circuit using Device 9

```
%% SSCB MOSFET Efficiency
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\IGBT\Infineon\IKW15N120BH6\SwitchEfficiency\SSCB_bi_Inf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\IGBT\Infineon\IGBT_1200V_TRENCHSTOP_IGBT6_L1_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\IGBT\Infineon\']; %data file path
circuit_filepath = [data_filepath 'IKW15N120BH6\SwitchEfficiency\SSCB_bi_Inf
    ']; %circuit simulation file path
component_filepath = [data_filepath 'IGBT_1200V_TRENCHSTOP_IGBT6_L1_temp.lib
    ']; %switch model file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
```

```
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = 0; %initial gate voltage
VgateF_vec = 15; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 29 *1e-9; %rise time
VgateTf_vec = 63 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 22; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_bi_mat = zeros(284,3); %initializes switch efficiency matrix
Row = 1; %intitialize row count for data dump

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                  {Vin
    }';
line{03} = 'XU1           input     gate_U1   TIE
    IKW15N120BH6_L1 TJ={VTj}';
line{04} = 'Rgate_U1      gate_U1   Vg1                                {
    Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{06} = 'XU2           output    gate_U2   TIE
    IKW15N120BH6_L1 TJ={VTj}';
line{07} = 'Rgate_U2      gate_U2   Vg2                                {
    Rgate}';
line{08} = 'Vgate_U2      Vg2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{09} = 'Rload         output    0                                  {
    Rload}';

line{10} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop) ' ' num2str(Tstart)
    ' ' num2str(Tmaxstep)];
line{11} = ['.lib ' component_filepath];
line{12} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL4=500 SRCSTEPS=100';
line{13} = '.ic V(input)={Vin} V(Vg1)={VgI} V(Vg2)={VgI}';
line{14} = ['.param Vin = ' num2str(Vin_vec)];
line{15} = ['.param VgI = ' num2str(VgateI_vec)];
line{16} = ['.param VgF = ' num2str(VgateF_vec)];
line{17} = ['.param VgD = ' num2str(VgateD_vec)];
line{18} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{19} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{20} = ['.param VgO = ' num2str(VgateO_vec)];
line{21} = ['.param Rgate = ' num2str(Rgate_vec)];
```

```
line{22} = ['.step param Rload ' Rload_vec];
line{23} = ['.step param VTj list ' num2str(Tj_vec)];
line{24} = '.backanno';
line{25} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(13,target));
            Pin = Vin.*abs(Iin);

            Vc_U1 = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Ve_U1 = nanmean(raw_data.variable_mat(2,target)); %same as Vtie
            Vce_U1 = Vc_U1-Ve_U1;
            Pce_U1 = Vce_U1.*abs(Iin);

            Vc_U2 = nanmean(raw_data.variable_mat(5,target)); %same as Vout
            Ve_U2 = nanmean(raw_data.variable_mat(2,target)); %same as Vtie
            Vec_U2 = Ve_U2-Vc_U2;
            Pec_U2 = Vec_U2.*abs(Iin);

            Efficiency = 100 * (Pin-(Pce_U1+Pec_U2))./Pin;

%             Tj = nanmean(raw_data.variable_mat(4,target)); %per doc VTj =
    Tj

            SWeff_bi_mat(Row,1) = abs(Iin);
%             SWeff_bi_mat(Row,2) = Tj;
            SWeff_bi_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

```

```
SWeff_bi_mat(1:71,2) = Tj_vec(1);
SWeff_bi_mat(72:142,2) = Tj_vec(2);
SWeff_bi_mat(143:213,2) = Tj_vec(3);
SWeff_bi_mat(214:284,2) = Tj_vec(4);

%% Data Save
save([data_filepath 'IKW15N120BH6\SwitchEfficiency\SWeff_bi_mat_temp.mat'],'
    SWeff_bi_mat');
```

Software C.21: Efficiency Simulation of a Bidirectional circuit using Device 10

```
%% SSCB MOSFET Efficiency
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\IGBT\Infineon\IKW15N120T2\SwitchEfficiency\SSCB_bi_Inf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\IGBT\Infineon\IGBT_1200V_TRENCHSTOP_IGBT6_L1_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\IGBT\Infineon\']; %data file path
circuit_filepath = [data_filepath 'IKW15N120T2\SwitchEfficiency\SSCB_bi_Inf'
    ]; %circuit simulation file path
component_filepath = [data_filepath 'IGBT_1200V_TRENCHSTOP2_L1_temp.lib']; %
    switch model file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = 0; %initial gate voltage
VgateF_vec = 15; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 25 *1e-9; %rise time
VgateTf_vec = 95 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 41.8; %settings from datasheet

Rload_vec = '30 100 1'; %sets load resistance row vector

Tj_vec = [-55 25 150 175]; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

span = 1; %initialize average start
target = zeros(1,1); %initializes target time matrix
SWeff_bi_mat = zeros(284,3); %initializes switch efficiency matrix
```

```
Row = 1; %intitialize row count for data dump

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                {Vin
    }';
line{03} = 'XU1           input     gate_U1   TIE
    IKW15N120T2_L1 TJ={VTj}';
line{04} = 'Rgate_U1      gate_U1   Vg1                              {
    Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{06} = 'XU2           output    gate_U2   TIE
    IKW15N120T2_L1 TJ={VTj}';
line{07} = 'Rgate_U2      gate_U2   Vg2                              {
    Rgate}';
line{08} = 'Vgate_U2      Vg2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{09} = 'Rload         output    0                                {
    Rload}';

line{10} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop) ' ' num2str(Tstart)
    ' ' num2str(Tmaxstep)];
line{11} = ['.lib ' component_filepath];
line{12} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL4=500 SRCSTEPS=100';
line{13} = '.ic V(input)={Vin} V(Vg1)={VgI} V(Vg2)={VgI}';
line{14} = ['.param Vin = ' num2str(Vin_vec)];
line{15} = ['.param VgI = ' num2str(VgateI_vec)];
line{16} = ['.param VgF = ' num2str(VgateF_vec)];
line{17} = ['.param VgD = ' num2str(VgateD_vec)];
line{18} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{19} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{20} = ['.param VgO = ' num2str(VgateO_vec)];
line{21} = ['.param Rgate = ' num2str(Rgate_vec)];
line{22} = ['.step param Rload ' Rload_vec];
line{23} = ['.step param VTj list ' num2str(Tj_vec)];
line{24} = '.backanno';
line{25} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);
```

```

%BiDirectional Circuit
time = raw_data.time_vect;
for ii = 1:size(time,2)
    if time(ii) >= AVGstart
        target(span) = ii;
        span = span+1;
        if time(ii) == Tstop
            Vin = nanmean(raw_data.variable_mat(1,target));
            Iin = nanmean(raw_data.variable_mat(13,target));
            Pin = Vin.*abs(Iin);

            Vc_U1 = nanmean(raw_data.variable_mat(1,target)); %same as Vin
            Ve_U1 = nanmean(raw_data.variable_mat(2,target)); %same as Vtie
            Vce_U1 = Vc_U1-Ve_U1;
            Pce_U1 = Vce_U1.*abs(Iin);

            Vc_U2 = nanmean(raw_data.variable_mat(5,target)); %same as Vout
            Ve_U2 = nanmean(raw_data.variable_mat(2,target)); %same as Vtie
            Vec_U2 = Ve_U2-Vc_U2;
            Pec_U2 = Vec_U2.*abs(Iin);

            Efficiency = 100 * (Pin-(Pce_U1+Pec_U2))./Pin;

%             Tj = nanmean(raw_data.variable_mat(4,target)); %per doc VTj =
    Tj

            SWeff_bi_mat(Row,1) = abs(Iin);
%             SWeff_bi_mat(Row,2) = Tj;
            SWeff_bi_mat(Row,3) = Efficiency;

            Row = Row+1; %increments to next row for data dump
            clear target
            span = 1;
        end
    end
end

SWeff_bi_mat(1:71,2) = Tj_vec(1);
SWeff_bi_mat(72:142,2) = Tj_vec(2);
SWeff_bi_mat(143:213,2) = Tj_vec(3);
SWeff_bi_mat(214:284,2) = Tj_vec(4);

%% Data Save
save([data_filepath 'IKW15N120T2\SwitchEfficiency\SWeff_bi_mat_temp.mat'],'
    SWeff_bi_mat');
```

Software C.22: Efficiency Simulation of a Bidirectional circuit using Device 11

```

%% RSA value
clc
close all
clear variables

%% Setup
% T_{j} & = T_{a} + P_{diss}(R_{\theta_{jc}} + R_{\theta_{cs}} + R_{\theta_{
    sa}})

```

```
Id = 1000/30;
Id_lim = 1000/34;
Tj = 175;
Tj_lim = 150;
Ta = 25;
TO247 = 0.005391;
SOT227 = 0.007091;

%% IMW120R060M1H
Rds = 60 *1e-3;
Pdiss = Id^2*Rds;
Pdiss_lim = Id_lim^2*Rds;
Rjc = 1;
Rcs = TO247;

syms Rsa
eqn1 = Ta + Pdiss*(Rjc + Rcs + Rsa) == Tj;
Rsa_sol = solve(eqn1,Rsa);

eqn2 = Ta + Pdiss_lim*(Rjc + Rcs + Rsa) == Tj_lim;
Rsa_lim_sol = solve(eqn2,Rsa);

Rsa_IMW120R060M1H = double(Rsa_sol);
Rsa_lim_IMW120R060M1H = double(Rsa_lim_sol);

%% LSIC1MO120E0080
Rds = 80 *1e-3;
Pdiss = Id^2*Rds;
Pdiss_lim = Id_lim^2*Rds;
Rjc = 0.7;
Rcs = TO247;

syms Rsa
eqn1 = Ta + Pdiss*(Rjc + Rcs + Rsa) == Tj;
Rsa_sol = solve(eqn1,Rsa);

eqn2 = Ta + Pdiss_lim*(Rjc + Rcs + Rsa) == Tj_lim;
Rsa_lim_sol = solve(eqn2,Rsa);

Rsa_LSIC1MO120E0080 = double(Rsa_sol);
Rsa_lim_LSIC1MO120E0080 = double(Rsa_lim_sol);

%% MSC080SMA120J
Rds = 80 *1e-3;
Pdiss = Id^2*Rds;
Pdiss_lim = Id_lim^2*Rds;
Rjc = 0.75;
Rcs = SOT227;

syms Rsa
eqn1 = Ta + Pdiss*(Rjc + Rcs + Rsa) == Tj;
Rsa_sol = solve(eqn1,Rsa);

eqn2 = Ta + Pdiss_lim*(Rjc + Rcs + Rsa) == Tj_lim;
Rsa_lim_sol = solve(eqn2,Rsa);

Rsa_MSC080SMA120J = double(Rsa_sol);
Rsa_lim_MSC080SMA120J = double(Rsa_lim_sol);

```

```
%% SCT3080KL
Rds = 80 *1e-3;
Pdiss = Id^2*Rds;
Pdiss_lim = Id_lim^2*Rds;
Rjc = 0.91;
Rcs = TO247;

syms Rsa
eqn1 = Ta + Pdiss*(Rjc + Rcs + Rsa) == Tj;
Rsa_sol = solve(eqn1,Rsa);

eqn2 = Ta + Pdiss_lim*(Rjc + Rcs + Rsa) == Tj_lim;
Rsa_lim_sol = solve(eqn2,Rsa);

Rsa_SCT3080KL = double(Rsa_sol);
Rsa_lim_SCT3080KL = double(Rsa_lim_sol);

%% UJ3N120070K3S
Rds = 70.5 *1e-3;
Pdiss = Id^2*Rds;
Pdiss_lim = Id_lim^2*Rds;
Rjc = 0.59;
Rcs = TO247;

syms Rsa
eqn1 = Ta + Pdiss*(Rjc + Rcs + Rsa) == Tj;
Rsa_sol = solve(eqn1,Rsa);

eqn2 = Ta + Pdiss_lim*(Rjc + Rcs + Rsa) == Tj_lim;
Rsa_lim_sol = solve(eqn2,Rsa);

Rsa_UJ3N120070K3S = double(Rsa_sol);
Rsa_lim_UJ3N120070K3S = double(Rsa_lim_sol);

%% UJ3C120080K3S
Rds = 80 *1e-3;
Pdiss = Id^2*Rds;
Pdiss_lim = Id_lim^2*Rds;
Rjc = 0.59;
Rcs = TO247;

syms Rsa
eqn1 = Ta + Pdiss*(Rjc + Rcs + Rsa) == Tj;
Rsa_sol = solve(eqn1,Rsa);

eqn2 = Ta + Pdiss_lim*(Rjc + Rcs + Rsa) == Tj_lim;
Rsa_lim_sol = solve(eqn2,Rsa);

Rsa_UJ3C120080K3S = double(Rsa_sol);
Rsa_lim_UJ3C120080K3S = double(Rsa_lim_sol);

%% UJ3C120070K3S
Rds = 70 *1e-3;
Pdiss = Id^2*Rds;
Pdiss_lim = Id_lim^2*Rds;
Rjc = 0.59;
Rcs = TO247;

syms Rsa
```

```
eqn1 = Ta + Pdiss*(Rjc + Rcs + Rsa) == Tj;
Rsa_sol = solve(eqn1,Rsa);

eqn2 = Ta + Pdiss_lim*(Rjc + Rcs + Rsa) == Tj_lim;
Rsa_lim_sol = solve(eqn2,Rsa);

Rsa_UJ3C120070K3S = double(Rsa_sol);
Rsa_lim_UJ3C120070K3S = double(Rsa_lim_sol);

%% C2M0080120D
Rds = 80 *1e-3;
Pdiss = Id^2*Rds;
Pdiss_lim = Id_lim^2*Rds;
Rjc = 0.65;
Rcs = TO247;

syms Rsa
eqn1 = Ta + Pdiss*(Rjc + Rcs + Rsa) == Tj;
Rsa_sol = solve(eqn1,Rsa);

eqn2 = Ta + Pdiss_lim*(Rjc + Rcs + Rsa) == Tj_lim;
Rsa_lim_sol = solve(eqn2,Rsa);

Rsa_C2M0080120D = double(Rsa_sol);
Rsa_lim_C2M0080120D = double(Rsa_lim_sol);

%% C3M0075120D
Rds = 75 *1e-3;
Pdiss = Id^2*Rds;
Pdiss_lim = Id_lim^2*Rds;
Rjc = 1.1;
Rcs = TO247;

syms Rsa
eqn1 = Ta + Pdiss*(Rjc + Rcs + Rsa) == Tj;
Rsa_sol = solve(eqn1,Rsa);

eqn2 = Ta + Pdiss_lim*(Rjc + Rcs + Rsa) == Tj_lim;
Rsa_lim_sol = solve(eqn2,Rsa);

Rsa_C3M0075120D = double(Rsa_sol);
Rsa_lim_C3M0075120D = double(Rsa_lim_sol);

%% IKW15N120BH6
Rds = 91.7 *1e-3; %estimated, Vge=15V, Ic=60, Vce=5.5
Pdiss = Id^2*Rds;
Pdiss_lim = Id_lim^2*Rds;
Rjc = 0.74;
Rcs = TO247;

syms Rsa
eqn1 = Ta + Pdiss*(Rjc + Rcs + Rsa) == Tj;
Rsa_sol = solve(eqn1,Rsa);

eqn2 = Ta + Pdiss_lim*(Rjc + Rcs + Rsa) == Tj_lim;
Rsa_lim_sol = solve(eqn2,Rsa);

Rsa_IKW15N120BH6 = double(Rsa_sol);
Rsa_lim_IKW15N120BH6 = double(Rsa_lim_sol);
```

```

%% IKW15N120T2
Rds = 79.2 *1e-3; %estimated, Vge=15V, Ic=60, Vce=4.75
Pdiss = Id^2*Rds;
Pdiss_lim = Id_lim^2*Rds;
Rjc = 0.63;
Rcs = TO247;

syms Rsa
eqn1 = Ta + Pdiss*(Rjc + Rcs + Rsa) == Tj;
Rsa_sol = solve(eqn1,Rsa);

eqn2 = Ta + Pdiss_lim*(Rjc + Rcs + Rsa) == Tj_lim;
Rsa_lim_sol = solve(eqn2,Rsa);

Rsa_IKW15N120T2 = double(Rsa_sol);
Rsa_lim_IKW15N120T2 = double(Rsa_lim_sol);
```

Software C.23: Derivation of $R_{\theta,sa}$ values for each simulated model.

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\
    SSCB_uni_Wolf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\TVS\littelfuse_tvs_diode_ak10_xxxc_b_spice_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_uni_Wolf_AK10170C']; %circuit
    simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.
    lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\
    littelfuse_tvs_diode_ak10_xxxc_b_spice_temp.lib']; %EDC file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
```

```
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rdrain_vec = 1 *1e-3;
Rgate_vec = 2.5; %settings from datasheet
Rsource_vec = 1 *1e-3;

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value /
    file with values
line{02} = 'Vin             input  0                             {Vin}';
line{03} = 'XU1             drain  gate   source Tj     Tc       C2M0080120D
    ';
line{04} = 'Rdrain          drain  input                         {Rdrain}';
line{05} = 'Rgate           gate   Vg                            {Rgate}';
line{06} = 'Rsource         output source                        {Rsource}';
line{07} = 'Vgate           Vg     output                        PULSE({VgI}
     {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{08} = 'Lload           output Load                          {Lload}';
line{09} = 'Rload           Load   0                             {Rload}';
line{10} = 'VTj             Tj     0                             {VTj}';
line{11} = 'VTc             Tc     0                             {VTc}';
line{12} = 'XU3             input  Stage1                        AK10-170C';
line{13} = 'XU3             Stage1 Stage2                        AK10-170C';
line{14} = 'XU3             Stage2 Stage3                        AK10-170C';
line{15} = 'XU3             Stage3 Stage4                        AK10-170C';
line{16} = 'XU3             Stage4 Stage5                        AK10-170C';
line{17} = 'XU3             Stage5 Stage6                        AK10-170C';
line{18} = 'Redc            Stage6 output                        1n';

line{19} = 'B1              Pedc   0                             V=(V(input)
    -V(Stage3))*I(Redc)';
line{20} = 'Ledc            Pedc   0                             1';

line{21} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep) ' startup'];
line{22} = ['.lib ' component_filepath];
line{23} = ['.lib ' EDC_filepath];
line{24} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{25} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg)={VgI}';
line{26} = ['.param Vin = ' num2str(Vin_vec)];
line{27} = ['.param VgI = ' num2str(VgateI_vec)];
line{28} = ['.param VgF = ' num2str(VgateF_vec)];
line{29} = ['.param VgD = ' num2str(VgateD_vec)];
line{30} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{31} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{32} = ['.param VgO = ' num2str(VgateO_vec)];
line{33} = ['.param Rdrain = ' num2str(Rdrain_vec)];
line{34} = ['.param Rgate = ' num2str(Rgate_vec)];
```

```
line{35} = ['.param Rsource = ' num2str(Rsource_vec)];
line{36} = ['.param Rload = ' num2str(Rload_vec)];
line{37} = ['.param Lload = ' Lload_vec];
line{38} = ['.param VTj = ' num2str(Tj_vec)];
line{39} = ['.param VTc = ' num2str(Tc_vec)];
line{40} = '.backanno';
line{41} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be
    removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <0.9999995);
[stp] = find(time(1,:) >1.0000025);

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(3,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage6 = raw_data.variable_mat(11,:);
Vstage6(:,col) = [];
Vstage6(:,[strt stp]) = [];

Vedc = Vinput - Vstage6;

Pedc = raw_data.variable_mat(10,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

Eedc = raw_data.variable_mat(18,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];

EDC_uni_mat(:,1) = time';
EDC_uni_mat(:,2) = Vedc';
EDC_uni_mat(:,3) = Pedc';
EDC_uni_mat(:,4) = (max(Eedc)-Eedc)';
```

```
139
140 %% Data Save
141 save([data_filepath '\EDC_AK10170C_temp.mat'],'EDC_uni_mat');
```

Software C.24: TVS Diode Unidirectional Performance Values of TVS 1

```
1  %% TVS Performance
2  clc
3  close all
4  clear variables
5
6  %% Paths
7  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\
       SSCB_uni_Wolf.cir
8  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
9  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\TVS\littelfuse_tvs_diode_ak10_xxxc_b_spice_temp.lib
10 %C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe
11
12 %% Setup
13 parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
       Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
14 data_filepath = [parent_filepath '\TVS\']; %data file path
15 circuit_filepath = [data_filepath 'SSCB_uni_Wolf_AK10380C']; %circuit
       simulation file path
16 component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.
       lib']; %switch model file path
17 EDC_filepath = [parent_filepath '\TVS\
       littelfuse_tvs_diode_ak10_xxxc_b_spice_temp.lib']; %EDC file path
18
19 %all parameters are scriptable, use .step param or .step param XXX list
20 Tprint = 0;
21 Tstop = 1.5; %transient analysis stop time
22 Tstart = 0;
23 Tmaxstep = 1 *1e-3; %transient analysis step
24 AVGstart = 0.9; %start averaging
25
26 Vin_vec = 1000; %input voltage
27
28 VgateI_vec = -5; %initial gate voltage
29 VgateF_vec = 20; %final gate voltage
30 VgateD_vec = 0.5; %gate voltage delay
31 VgateTr_vec = 22 *1e-9; %rise time
32 VgateTf_vec = 14 *1e-9; %fall time
33 VgateO_vec = 0.5; %gate voltage ON time
34
35 Rdrain_vec = 1 *1e-3;
36 Rgate_vec = 2.5; %settings from datasheet
37 Rsource_vec = 1 *1e-3;
38
39 Rload_vec = 30; %sets load resistance row vector
40 Lload_vec = '100u'; %sets load inductance row vector
41
42 Tj_vec = 25; %sets Tj row vector
43 Tc_vec = 25; %sets Tc row vector
44
```

```
%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value / file with values
line{02} = 'Vin             input  0                             {Vin}';
line{03} = 'XU1             drain  gate   source Tj     Tc       C2M0080120D';
line{04} = 'Rdrain          drain  input                         {Rdrain}';
line{05} = 'Rgate           gate   Vg                            {Rgate}';
line{06} = 'Rsource         output source                        {Rsource}';
line{07} = 'Vgate           Vg     output                        PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{08} = 'Lload           output Load                          {Lload}';
line{09} = 'Rload           Load   0                             {Rload}';
line{10} = 'VTj             Tj     0                             {VTj}';
line{11} = 'VTc             Tc     0                             {VTc}';
line{12} = 'XU3             input  Stage1                        AK10-380C';
line{13} = 'XU3             Stage1 Stage2                        AK10-380C';
line{14} = 'XU3             Stage2 Stage3                        AK10-380C';
line{15} = 'Redc            Stage3 output                        1n';

line{16} = 'B1              Pedc   0                             V=(V(input)-V(Stage3))*I(Redc)';
line{17} = 'Ledc            Pedc   0                             1';

line{18} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart) ' ' num2str(Tmaxstep) ' startup'];
line{19} = ['.lib ' component_filepath];
line{20} = ['.lib ' EDC_filepath];
line{21} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e-4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{22} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg)={VgI}';
line{23} = ['.param Vin = ' num2str(Vin_vec)];
line{24} = ['.param VgI = ' num2str(VgateI_vec)];
line{25} = ['.param VgF = ' num2str(VgateF_vec)];
line{26} = ['.param VgD = ' num2str(VgateD_vec)];
line{27} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{28} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{29} = ['.param VgO = ' num2str(VgateO_vec)];
line{30} = ['.param Rdrain = ' num2str(Rdrain_vec)];
line{31} = ['.param Rgate = ' num2str(Rgate_vec)];
line{32} = ['.param Rsource = ' num2str(Rsource_vec)];
line{33} = ['.param Rload = ' num2str(Rload_vec)];
line{34} = ['.param Lload = ' Lload_vec];
line{35} = ['.param VTj = ' num2str(Tj_vec)];
line{36} = ['.param VTc = ' num2str(Tc_vec)];
line{37} = '.backanno';
line{38} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
```

```
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be
    removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <0.9999995);
[stp] = find(time(1,:) >1.0000025);

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(3,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage3 = raw_data.variable_mat(11,:);
Vstage3(:,col) = [];
Vstage3(:,[strt stp]) = [];

Vedc = Vinput - Vstage3;

Pedc = raw_data.variable_mat(10,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

Eedc = raw_data.variable_mat(15,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];

EDC_uni_mat(:,1) = time';
EDC_uni_mat(:,2) = Vedc';
EDC_uni_mat(:,3) = Pedc';
EDC_uni_mat(:,4) = (max(Eedc)-Eedc)';

%% Data Save
save([data_filepath '\EDC_AK10380C_temp.mat'],'EDC_uni_mat');
```

Software C.25: TVS Diode Unidirectional Performance Values of TVS 2

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\
    SSCB_uni_Wolf.cir
```

```
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\TVS\littelfuse_tvs_diode_ak10_xxxc_b_spice_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_uni_Wolf_AK10530C']; %circuit simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\littelfuse_tvs_diode_ak10_xxxc_b_spice_temp.lib']; %EDC file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rdrain_vec = 1 *1e-3;
Rgate_vec = 2.5; %settings from datasheet
Rsource_vec = 1 *1e-3;

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value / file with values
line{02} = 'Vin             input  0                             {Vin}';
line{03} = 'XU1             drain  gate   source Tj     Tc       C2M0080120D';
line{04} = 'Rdrain          drain  input                         {Rdrain}';
line{05} = 'Rgate           gate   Vg                            {Rgate}';
line{06} = 'Rsource         output source                        {Rsource}';
line{07} = 'Vgate           Vg     output                        PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{08} = 'Lload           output Load                          {Lload}';
line{09} = 'Rload           Load   0                             {Rload}';
line{10} = 'VTj             Tj     0                             {VTj}';
```

```
line{11} = 'VTc             Tc      0                               {VTc}';
line{12} = 'XU3             input   Stage1                          AK10-530C';
line{13} = 'XU3             Stage1  Stage2                          AK10-530C';
line{14} = 'Redc            Stage2  output                          1n';

line{15} = 'B1              Pedc    0                               V=(V(input)
    -V(Stage3))*I(Redc)';
line{16} = 'Ledc            Pedc    0                               1';

line{17} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep) ' startup'];
line{18} = ['.lib ' component_filepath];
line{19} = ['.lib ' EDC_filepath];
line{20} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{21} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg)={VgI}';
line{22} = ['.param Vin = ' num2str(Vin_vec)];
line{23} = ['.param VgI = ' num2str(VgateI_vec)];
line{24} = ['.param VgF = ' num2str(VgateF_vec)];
line{25} = ['.param VgD = ' num2str(VgateD_vec)];
line{26} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{27} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{28} = ['.param VgO = ' num2str(VgateO_vec)];
line{29} = ['.param Rdrain = ' num2str(Rdrain_vec)];
line{30} = ['.param Rgate = ' num2str(Rgate_vec)];
line{31} = ['.param Rsource = ' num2str(Rsource_vec)];
line{32} = ['.param Rload = ' num2str(Rload_vec)];
line{33} = ['.param Lload = ' Lload_vec];
line{34} = ['.param VTj = ' num2str(Tj_vec)];
line{35} = ['.param VTc = ' num2str(Tc_vec)];
line{36} = '.backanno';
line{37} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be
    removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <0.9999995);
[stp] = find(time(1,:) >1.0000025);
```

```

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(3,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage2 = raw_data.variable_mat(11,:);
Vstage2(:,col) = [];
Vstage2(:,[strt stp]) = [];

Vedc = Vinput - Vstage2;

Pedc = raw_data.variable_mat(10,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

Eedc = raw_data.variable_mat(14,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];

EDC_uni_mat(:,1) = time';
EDC_uni_mat(:,2) = Vedc';
EDC_uni_mat(:,3) = Pedc';
EDC_uni_mat(:,4) = (max(Eedc)-Eedc)';

%% Data Save
save([data_filepath '\EDC_AK10530C_temp.mat'],'EDC_uni_mat');
```

Software C.26: TVS Diode Unidirectional Performance Values of TVS 3

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\
    SSCB_uni_Wolf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\TVS\littelfuse_tvs_diode_ak3_xxxc_b_spice_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_uni_Wolf_AK3380C']; %circuit
    simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.
    lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\
    littelfuse_tvs_diode_ak3_xxxc_b_spice_temp.lib']; %EDC file path

%all parameters are scriptable, use .step param or .step param XXX list
```

```
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rdrain_vec = 1 *1e-3;
Rgate_vec = 2.5; %settings from datasheet
Rsource_vec = 1 *1e-3;

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value / file with values
line{02} = 'Vin             input  0                             {Vin}';
line{03} = 'XU1             drain  gate   source Tj     Tc       C2M0080120D';
line{04} = 'Rdrain          drain  input                         {Rdrain}';
line{05} = 'Rgate           gate   Vg                            {Rgate}';
line{06} = 'Rsource         output source                        {Rsource}';
line{07} = 'Vgate           Vg     output                        PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{08} = 'Lload           output Load                          {Lload}';
line{09} = 'Rload           Load   0                             {Rload}';
line{10} = 'VTj             Tj     0                             {VTj}';
line{11} = 'VTc             Tc     0                             {VTc}';
line{12} = 'XU3             input  Stage1                        AK3-380C';
line{13} = 'XU3             Stage1 Stage2                        AK3-380C';
line{14} = 'XU3             Stage2 Stage3                        AK3-380C';
line{15} = 'Redc            Stage3 output                        1n';

line{16} = 'B1              Pedc   0                             V=(V(input)-V(Stage3))*I(Redc)';
line{17} = 'Ledc            Pedc   0                             1';

line{18} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart) ' ' num2str(Tmaxstep) ' startup'];
line{19} = ['.lib ' component_filepath];
line{20} = ['.lib ' EDC_filepath];
line{21} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e-4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{22} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg)={VgI}';
line{23} = ['.param Vin = ' num2str(Vin_vec)];
```

```
line{24} = ['.param VgI = ' num2str(VgateI_vec)];
line{25} = ['.param VgF = ' num2str(VgateF_vec)];
line{26} = ['.param VgD = ' num2str(VgateD_vec)];
line{27} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{28} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{29} = ['.param VgO = ' num2str(VgateO_vec)];
line{30} = ['.param Rdrain = ' num2str(Rdrain_vec)];
line{31} = ['.param Rgate = ' num2str(Rgate_vec)];
line{32} = ['.param Rsource = ' num2str(Rsource_vec)];
line{33} = ['.param Rload = ' num2str(Rload_vec)];
line{34} = ['.param Lload = ' Lload_vec];
line{35} = ['.param VTj = ' num2str(Tj_vec)];
line{36} = ['.param VTc = ' num2str(Tc_vec)];
line{37} = '.backanno';
line{38} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be
    removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <0.9999995);
[stp] = find(time(1,:) >1.0000025);

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(3,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage3 = raw_data.variable_mat(11,:);
Vstage3(:,col) = [];
Vstage3(:,[strt stp]) = [];

Vedc = Vinput - Vstage3;

Pedc = raw_data.variable_mat(10,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

```

```
Eedc = raw_data.variable_mat(15,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];

EDC_uni_mat(:,1) = time';
EDC_uni_mat(:,2) = Vedc';
EDC_uni_mat(:,3) = Pedc';
EDC_uni_mat(:,4) = (max(Eedc)-Eedc)';

%% Data Save
save([data_filepath '\EDC_AK3380C_temp.mat'],'EDC_uni_mat');
```

Software C.27: TVS Diode Unidirectional Performance Values of TVS 4

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\
    SSCB_uni_Wolf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\TVS\littelfuse_tvs_diode_ak6_xxxc_b_spice_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_uni_Wolf_AK6170C']; %circuit
    simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.
    lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\
    littelfuse_tvs_diode_ak6_xxxc_b_spice_temp.lib']; %EDC file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rdrain_vec = 1 *1e-3;
Rgate_vec = 2.5; %settings from datasheet
```

```
Rsource_vec = 1 *1e-3;

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1   node2   node3   node4   node5    value / file with values
line{02} = 'Vin             input   0                                {Vin}';
line{03} = 'XU1             drain   gate    source Tj       Tc       C2M0080120D';
line{04} = 'Rdrain          drain   input                            {Rdrain}';
line{05} = 'Rgate           gate    Vg                               {Rgate}';
line{06} = 'Rsource         output  source                           {Rsource}';
line{07} = 'Vgate           Vg      output                           PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{08} = 'Lload           output  Load                             {Lload}';
line{09} = 'Rload           Load    0                                {Rload}';
line{10} = 'VTj             Tj      0                                {VTj}';
line{11} = 'VTc             Tc      0                                {VTc}';
line{12} = 'XU3             input   Stage1                           AK6-170C';
line{13} = 'XU3             Stage1  Stage2                           AK6-170C';
line{14} = 'XU3             Stage2  Stage3                           AK6-170C';
line{15} = 'XU3             Stage3  Stage4                           AK6-170C';
line{16} = 'XU3             Stage4  Stage5                           AK6-170C';
line{17} = 'XU3             Stage5  Stage6                           AK6-170C';
line{18} = 'Redc            Stage6  output                           1n';

line{19} = 'B1              Pedc    0                                V=(V(input)-V(Stage3))*I(Redc)';
line{20} = 'Ledc            Pedc    0                                1';

line{21} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart) ' ' num2str(Tmaxstep) ' startup'];
line{22} = ['.lib ' component_filepath];
line{23} = ['.lib ' EDC_filepath];
line{24} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e-4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{25} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg)={VgI}';
line{26} = ['.param Vin = ' num2str(Vin_vec)];
line{27} = ['.param VgI = ' num2str(VgateI_vec)];
line{28} = ['.param VgF = ' num2str(VgateF_vec)];
line{29} = ['.param VgD = ' num2str(VgateD_vec)];
line{30} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{31} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{32} = ['.param VgO = ' num2str(VgateO_vec)];
line{33} = ['.param Rdrain = ' num2str(Rdrain_vec)];
line{34} = ['.param Rgate = ' num2str(Rgate_vec)];
line{35} = ['.param Rsource = ' num2str(Rsource_vec)];
line{36} = ['.param Rload = ' num2str(Rload_vec)];
line{37} = ['.param Lload = ' Lload_vec];
line{38} = ['.param VTj = ' num2str(Tj_vec)];
line{39} = ['.param VTc = ' num2str(Tc_vec)];
line{40} = '.backanno';
```

```
line{41} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be
    removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <0.9999995);
[stp] = find(time(1,:) >1.0000025);

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(3,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage6 = raw_data.variable_mat(11,:);
Vstage6(:,col) = [];
Vstage6(:,[strt stp]) = [];

Vedc = Vinput - Vstage6;

Pedc = raw_data.variable_mat(10,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

Eedc = raw_data.variable_mat(18,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];

EDC_uni_mat(:,1) = time';
EDC_uni_mat(:,2) = Vedc';
EDC_uni_mat(:,3) = Pedc';
EDC_uni_mat(:,4) = (max(Eedc)-Eedc)';

%% Data Save
save([data_filepath '\EDC_AK6170C_temp.mat'],'EDC_uni_mat');
```

Software C.28: TVS Diode Unidirectional Performance Values of TVS 5

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\SSCB_uni_Wolf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\TVS\littelfuse_tvs_diode_ak6_xxxc_b_spice_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_uni_Wolf_AK6380C']; %circuit simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\littelfuse_tvs_diode_ak6_xxxc_b_spice_temp.lib']; %EDC file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rdrain_vec = 1 *1e-3;
Rgate_vec = 2.5; %settings from datasheet
Rsource_vec = 1 *1e-3;

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1   node2   node3   node4   node5    value / file with values
line{02} = 'Vin             input   0                               {Vin}';
```

```
line{03} = 'XU1             drain   gate    source Tj      Tc        C2M0080120D
    ';
line{04} = 'Rdrain          drain   input                            {Rdrain}';
line{05} = 'Rgate           gate    Vg                               {Rgate}';
line{06} = 'Rsource         output  source                           {Rsource}';
line{07} = 'Vgate           Vg      output                           PULSE({VgI}
    {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{08} = 'Lload           output  Load                             {Lload}';
line{09} = 'Rload           Load    0                                {Rload}';
line{10} = 'VTj             Tj      0                                {VTj}';
line{11} = 'VTc             Tc      0                                {VTc}';
line{12} = 'XU3             input   Stage1                           AK6-380C';
line{13} = 'XU3             Stage1  Stage2                           AK6-380C';
line{14} = 'XU3             Stage2  Stage3                           AK6-380C';
line{15} = 'Redc            Stage3  output                           1n';

line{16} = 'B1              Pedc    0                                V=(V(input)
    -V(Stage3))*I(Redc)';
line{17} = 'Ledc            Pedc    0                                1';

line{18} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
    ' ' num2str(Tmaxstep) ' startup'];
line{19} = ['.lib ' component_filepath];
line{20} = ['.lib ' EDC_filepath];
line{21} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{22} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg)={VgI}';
line{23} = ['.param Vin = ' num2str(Vin_vec)];
line{24} = ['.param VgI = ' num2str(VgateI_vec)];
line{25} = ['.param VgF = ' num2str(VgateF_vec)];
line{26} = ['.param VgD = ' num2str(VgateD_vec)];
line{27} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{28} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{29} = ['.param VgO = ' num2str(VgateO_vec)];
line{30} = ['.param Rdrain = ' num2str(Rdrain_vec)];
line{31} = ['.param Rgate = ' num2str(Rgate_vec)];
line{32} = ['.param Rsource = ' num2str(Rsource_vec)];
line{33} = ['.param Rload = ' num2str(Rload_vec)];
line{34} = ['.param Lload = ' Lload_vec];
line{35} = ['.param VTj = ' num2str(Tj_vec)];
line{36} = ['.param VTc = ' num2str(Tc_vec)];
line{37} = '.backanno';
line{38} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
```

```
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be
    removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <0.9999995);
[stp] = find(time(1,:) >1.0000025);

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(3,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage3 = raw_data.variable_mat(11,:);
Vstage3(:,col) = [];
Vstage3(:,[strt stp]) = [];

Vedc = Vinput - Vstage3;

Pedc = raw_data.variable_mat(10,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

Eedc = raw_data.variable_mat(15,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];

EDC_uni_mat(:,1) = time';
EDC_uni_mat(:,2) = Vedc';
EDC_uni_mat(:,3) = Pedc';
EDC_uni_mat(:,4) = (max(Eedc)-Eedc)';

%% Data Save
save([data_filepath '\EDC_AK6380C_temp.mat'],'EDC_uni_mat');
```

Software C.29: TVS Diode Unidirectional Performance Values of TVS 6

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\
    SSCB_uni_Wolf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\TVS\PTVS10-xxxC-TH_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
```

```
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_uni_Wolf_PTVS10']; %circuit simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\PTVS10-xxxC-TH_temp.lib']; %EDC file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rdrain_vec = 1 *1e-3;
Rgate_vec = 2.5; %settings from datasheet
Rsource_vec = 1 *1e-3;

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value / file with values
line{02} = 'Vin             input  0                             {Vin}';
line{03} = 'XU1             drain  gate   source Tj     Tc       C2M0080120D';
line{04} = 'Rdrain          drain  input                         {Rdrain}';
line{05} = 'Rgate           gate   Vg                            {Rgate}';
line{06} = 'Rsource         output source                        {Rsource}';
line{07} = 'Vgate           Vg     output                        PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{08} = 'Lload           output Load                          {Lload}';
line{09} = 'Rload           Load   0                             {Rload}';
line{10} = 'VTj             Tj     0                             {VTj}';
line{11} = 'VTc             Tc     0                             {VTc}';
line{12} = 'XU3             input  Stage1                        PTVS10-380C-TH';
line{13} = 'XU3             Stage1 Stage2                        PTVS10-380C-TH';
line{14} = 'XU3             Stage2 Stage3                        PTVS10-380C-TH';
```

```
line{15} = 'Redc            Stage3 output                          1n';

line{16} = 'B1              Pedc   0                               V=(V(input)-V(Stage3))*I(Redc)';
line{17} = 'Ledc            Pedc   0                               1';

line{18} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart) ' ' num2str(Tmaxstep) ' startup'];
line{19} = ['.lib ' component_filepath];
line{20} = ['.lib ' EDC_filepath];
line{21} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e-4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{22} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg)={VgI}';
line{23} = ['.param Vin = ' num2str(Vin_vec)];
line{24} = ['.param VgI = ' num2str(VgateI_vec)];
line{25} = ['.param VgF = ' num2str(VgateF_vec)];
line{26} = ['.param VgD = ' num2str(VgateD_vec)];
line{27} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{28} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{29} = ['.param VgO = ' num2str(VgateO_vec)];
line{30} = ['.param Rdrain = ' num2str(Rdrain_vec)];
line{31} = ['.param Rgate = ' num2str(Rgate_vec)];
line{32} = ['.param Rsource = ' num2str(Rsource_vec)];
line{33} = ['.param Rload = ' num2str(Rload_vec)];
line{34} = ['.param Lload = ' Lload_vec];
line{35} = ['.param VTj = ' num2str(Tj_vec)];
line{36} = ['.param VTc = ' num2str(Tc_vec)];
line{37} = '.backanno';
line{38} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "' circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. > In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <0.9999995);
[stp] = find(time(1,:) >1.0000025);

time(:,[strt stp]) = [];

```

```
Vinput = raw_data.variable_mat(3,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage3 = raw_data.variable_mat(11,:);
Vstage3(:,col) = [];
Vstage3(:,[strt stp]) = [];

Vedc = Vinput - Vstage3;

Pedc = raw_data.variable_mat(10,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

Eedc = raw_data.variable_mat(15,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];

EDC_uni_mat(:,1) = time';
EDC_uni_mat(:,2) = Vedc';
EDC_uni_mat(:,3) = Pedc';
EDC_uni_mat(:,4) = (max(Eedc)-Eedc)';

%% Data Save
save([data_filepath '\EDC_PTVS10_temp.mat'],'EDC_uni_mat');
```

Software C.30: TVS Diode Unidirectional Performance Values of TVS 7

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\
    SSCB_uni_Wolf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\TVS\PTVS3-xxxC-TH_HV_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_uni_Wolf_PTVS3']; %circuit
    simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.
    lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\PTVS3-xxxC-TH_HV_temp.lib']; %EDC file
     path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
```

```
Tmaxstep = 1 *1e-3; %transient analysis step
AVGstart = 0.9; %start averaging

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rdrain_vec = 1 *1e-3;
Rgate_vec = 2.5; %settings from datasheet
Rsource_vec = 1 *1e-3;

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%UniDirectional Circuit
%           name            node1  node2  node3  node4  node5    value /
    file with values
line{02} = 'Vin             input  0                             {Vin}';
line{03} = 'XU1             drain  gate   source Tj     Tc       C2M0080120D
    ';
line{04} = 'Rdrain          drain  input                         {Rdrain}';
line{05} = 'Rgate           gate   Vg                            {Rgate}';
line{06} = 'Rsource         output source                        {Rsource}';
line{07} = 'Vgate           Vg     output                        PULSE({VgI}
     {VgF} {VgD} {VgTr} {VgTf} {VgO})';
line{08} = 'Lload           output Load                          {Lload}';
line{09} = 'Rload           Load   0                             {Rload}';
line{10} = 'VTj             Tj     0                             {VTj}';
line{11} = 'VTc             Tc     0                             {VTc}';
line{12} = 'XU3             input  Stage1                        PTVS3-380C-
    TH';
line{13} = 'XU3             Stage1 Stage2                        PTVS3-380C-
    TH';
line{14} = 'XU3             Stage2 Stage3                        PTVS3-380C-
    TH';
line{15} = 'Redc            Stage3 output                        1n';

line{16} = 'B1              Pedc   0                             V=(V(input)
    -V(Stage3))*I(Redc)';
line{17} = 'Ledc            Pedc   0                             1';

line{18} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep) ' startup'];
line{19} = ['.lib ' component_filepath];
line{20} = ['.lib ' EDC_filepath];
line{21} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{22} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg)={VgI}';
line{23} = ['.param Vin = ' num2str(Vin_vec)];
```

```
line{24} = ['.param VgI = ' num2str(VgateI_vec)];
line{25} = ['.param VgF = ' num2str(VgateF_vec)];
line{26} = ['.param VgD = ' num2str(VgateD_vec)];
line{27} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{28} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{29} = ['.param VgO = ' num2str(VgateO_vec)];
line{30} = ['.param Rdrain = ' num2str(Rdrain_vec)];
line{31} = ['.param Rgate = ' num2str(Rgate_vec)];
line{32} = ['.param Rsource = ' num2str(Rsource_vec)];
line{33} = ['.param Rload = ' num2str(Rload_vec)];
line{34} = ['.param Lload = ' Lload_vec];
line{35} = ['.param VTj = ' num2str(Tj_vec)];
line{36} = ['.param VTc = ' num2str(Tc_vec)];
line{37} = '.backanno';
line{38} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%UniDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be
    removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <0.9999995);
[stp] = find(time(1,:) >1.0000025);

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(3,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage3 = raw_data.variable_mat(11,:);
Vstage3(:,col) = [];
Vstage3(:,[strt stp]) = [];

Vedc = Vinput - Vstage3;

Pedc = raw_data.variable_mat(10,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

```

```
Eedc = raw_data.variable_mat(15,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];

EDC_uni_mat(:,1) = time';
EDC_uni_mat(:,2) = Vedc';
EDC_uni_mat(:,3) = Pedc';
EDC_uni_mat(:,4) = (max(Eedc)-Eedc)';

%% Data Save
save([data_filepath '\EDC_PTVS3_temp.mat'],'EDC_uni_mat');
```

Software C.31: TVS Diode Unidirectional Performance Values of TVS 8

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\SSCB_bi_Wolf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\TVS\littelfuse_tvs_diode_ak10_xxxc_b_spice_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_bi_Wolf_AK10170C']; %circuit simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\littelfuse_tvs_diode_ak10_xxxc_b_spice_temp.lib']; %EDC file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step

Tsrt = 0.9999999; %start time for data collection
Tstp = 1.000015; %stop time for data collection

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

```

```
Rgate_vec = 2.5; %settings from datasheet

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                  {Vin
    }';
line{03} = 'XU1           input     gate_U1   TIE       Tj     Tc
    C2M0080120D';
line{04} = 'Rgate_U1      gate_U1   Vg1                                {
    Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{06} = 'XU2           output    gate_U2   TIE       Tj     Tc
    C2M0080120D';
line{07} = 'Rgate_U2      gate_U2   VG2                                {
    Rgate}';
line{08} = 'Vgate_U2      VG2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{09} = 'Lload         output    Load                               {
    Lload}';
line{10} = 'Rload         Load      0                                  {
    Rload}';
line{11} = 'VTj           Tj        0                                  {VTj
    }';
line{12} = 'VTc           Tc        0                                  {VTc
    }';

line{13} = 'XU3             input  Stage1                         AK10-170C';
line{14} = 'XU4             Stage1 Stage2                         AK10-170C';
line{15} = 'XU5             Stage2 Stage3                         AK10-170C';
line{16} = 'XU6             Stage3 Stage4                         AK10-170C';
line{17} = 'XU7             Stage4 Stage5                         AK10-170C';
line{18} = 'XU8             Stage5 Stage6                         AK10-170C';
line{19} = 'Redc            Stage6 output                         1n';

line{20} = 'B1              Pedc   0                              V=(V(input)
    -V(Stage6))*I(Redc)';
line{21} = 'Ledc            Pedc   0                              1';

line{22} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep) ' startup'];
line{23} = ['.lib ' component_filepath];
line{24} = ['.lib ' EDC_filepath];
line{25} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{26} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg1)={VgI} V(Vg2)={
    VgI}';
line{27} = ['.param Vin = ' num2str(Vin_vec)];
line{28} = ['.param VgI = ' num2str(VgateI_vec)];
line{29} = ['.param VgF = ' num2str(VgateF_vec)];
```

```
line{30} = ['.param VgD = ' num2str(VgateD_vec)];
line{31} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{32} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{33} = ['.param VgO = ' num2str(VgateO_vec)];
line{34} = ['.param Rgate = ' num2str(Rgate_vec)];
line{35} = ['.param Rload = ' num2str(Rload_vec)];
line{36} = ['.param Lload = ' Lload_vec];
line{37} = ['.param VTj = ' num2str(Tj_vec)];
line{38} = ['.param VTc = ' num2str(Tc_vec)];
line{39} = '.backanno';
line{40} = '.end';
%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be
    removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <Tsrt);
[stp] = find(time(1,:) >Tstp);

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(1,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage6 = raw_data.variable_mat(16,:);
Vstage6(:,col) = [];
Vstage6(:,[strt stp]) = [];

Vedc = Vinput - Vstage6;

Pedc = raw_data.variable_mat(17,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

Eedc = raw_data.variable_mat(19,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];

time = time-Tsrt; %sets the data to start at 0
```

```
time = time*1e6; %defines the time scale to be micro-seconds
Pedc = Pedc/1e3; %defines the power scale to be kilo-watts
Eedc = Eedc*1e3; %defines the energy scale to be mili-joules

EDC_bi_mat(:,1) = time';
EDC_bi_mat(:,2) = Vedc';
EDC_bi_mat(:,3) = Pedc';
EDC_bi_mat(:,4) = Eedc'-min(Eedc);

% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,2))
% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,3))
% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,4))

%% Data Save
save([data_filepath '\EDC_AK10170C_temp.mat'],'EDC_bi_mat');
```

Software C.32: TVS Diode Bidirectional Performance Values of TVS 1

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\SSCB_bi_Wolf
    .cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\TVS\littelfuse_tvs_diode_ak10_xxxc_b_spice_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_bi_Wolf_AK10380C']; %circuit
    simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.
    lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\
    littelfuse_tvs_diode_ak10_xxxc_b_spice_temp.lib']; %EDC file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step

Tsrt = 0.9999999; %start time for data collection
Tstp = 1.000015; %stop time for data collection

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
```

```
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 2.5; %settings from datasheet

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                    {Vin
    }';
line{03} = 'XU1           input     gate_U1   TIE       Tj     Tc
    C2M0080120D';
line{04} = 'Rgate_U1      gate_U1   Vg1                                  {
    Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{06} = 'XU2           output    gate_U2   TIE       Tj     Tc
    C2M0080120D';
line{07} = 'Rgate_U2      gate_U2   VG2                                  {
    Rgate}';
line{08} = 'Vgate_U2      VG2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{09} = 'Lload         output    Load                                 {
    Lload}';
line{10} = 'Rload         Load      0                                    {
    Rload}';
line{11} = 'VTj           Tj        0                                    {VTj
    }';
line{12} = 'VTc           Tc        0                                    {VTc
    }';

line{13} = 'XU3             input  Stage1                          AK10-380C';
line{14} = 'XU4             Stage1 Stage2                          AK10-380C';
line{15} = 'XU5             Stage2 Stage3                          AK10-380C';
line{16} = 'Redc            Stage3 output                          1n';

line{17} = 'B1              Pedc   0                              V=(V(input)
    -V(Stage3))*I(Redc)';
line{18} = 'Ledc            Pedc   0                              1';

line{19} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep) ' startup'];
line{20} = ['.lib ' component_filepath];
line{21} = ['.lib ' EDC_filepath];
line{22} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{23} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg1)={VgI} V(Vg2)={
    VgI}';
line{24} = ['.param Vin = ' num2str(Vin_vec)];
line{25} = ['.param VgI = ' num2str(VgateI_vec)];
```

```
line{26} = ['.param VgF = ' num2str(VgateF_vec)];
line{27} = ['.param VgD = ' num2str(VgateD_vec)];
line{28} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{29} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{30} = ['.param VgO = ' num2str(VgateO_vec)];
line{31} = ['.param Rgate = ' num2str(Rgate_vec)];
line{32} = ['.param Rload = ' num2str(Rload_vec)];
line{33} = ['.param Lload = ' Lload_vec];
line{34} = ['.param VTj = ' num2str(Tj_vec)];
line{35} = ['.param VTc = ' num2str(Tc_vec)];
line{36} = '.backanno';
line{37} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be
    removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <Tsrt);
[stp] = find(time(1,:) >Tstp);

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(1,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage3 = raw_data.variable_mat(13,:);
Vstage3(:,col) = [];
Vstage3(:,[strt stp]) = [];

Vedc = Vinput - Vstage3;

Pedc = raw_data.variable_mat(14,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

Eedc = raw_data.variable_mat(16,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];
```

```

time = time-Tsrt; %sets the data to start at 0
time = time*1e6; %defines the time scale to be micro-seconds
Pedc = Pedc/1e3; %defines the power scale to be kilo-watts
Eedc = Eedc*1e3; %defines the energy scale to be mili-joules

EDC_bi_mat(:,1) = time';
EDC_bi_mat(:,2) = Vedc';
EDC_bi_mat(:,3) = Pedc';
EDC_bi_mat(:,4) = Eedc'-min(Eedc);

% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,2))
% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,3))
% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,4))

%% Data Save
save([data_filepath '\EDC_AK10380C_temp.mat'],'EDC_bi_mat');
```

Software C.33: TVS Diode Bidirectional Performance Values of TVS 2

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\SSCB_bi_Wolf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\TVS\littelfuse_tvs_diode_ak10_xxxc_b_spice_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_bi_Wolf_AK10530C']; %circuit simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\littelfuse_tvs_diode_ak10_xxxc_b_spice_temp.lib']; %EDC file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step

Tsrt = 0.9999999; %start time for data collection
Tstp = 1.000015; %stop time for data collection

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
```

```
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 2.5; %settings from datasheet

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                    {Vin
    }';
line{03} = 'XU1           input     gate_U1   TIE       Tj     Tc
    C2M0080120D';
line{04} = 'Rgate_U1      gate_U1   Vg1                                  {
    Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{06} = 'XU2           output    gate_U2   TIE       Tj     Tc
    C2M0080120D';
line{07} = 'Rgate_U2      gate_U2   VG2                                  {
    Rgate}';
line{08} = 'Vgate_U2      VG2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{09} = 'Lload         output    Load                                 {
    Lload}';
line{10} = 'Rload         Load      0                                    {
    Rload}';
line{11} = 'VTj           Tj        0                                    {VTj
    }';
line{12} = 'VTc           Tc        0                                    {VTc
    }';

line{13} = 'XU3             input  Stage1                        AK10-530C';
line{14} = 'XU4             Stage1 Stage2                        AK10-530C';
line{15} = 'Redc            Stage2 output                        1n';

line{16} = 'B1              Pedc   0                             V=(V(input)
    -V(Stage2))*I(Redc)';
line{17} = 'Ledc            Pedc   0                             1';

line{18} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep) ' startup'];
line{19} = ['.lib ' component_filepath];
line{20} = ['.lib ' EDC_filepath];
line{21} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{22} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg1)={VgI} V(Vg2)={
    VgI}';
line{23} = ['.param Vin = ' num2str(Vin_vec)];
```

```
line{24} = ['.param VgI = ' num2str(VgateI_vec)];
line{25} = ['.param VgF = ' num2str(VgateF_vec)];
line{26} = ['.param VgD = ' num2str(VgateD_vec)];
line{27} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{28} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{29} = ['.param VgO = ' num2str(VgateO_vec)];
line{30} = ['.param Rgate = ' num2str(Rgate_vec)];
line{31} = ['.param Rload = ' num2str(Rload_vec)];
line{32} = ['.param Lload = ' Lload_vec];
line{33} = ['.param VTj = ' num2str(Tj_vec)];
line{34} = ['.param VTc = ' num2str(Tc_vec)];
line{35} = '.backanno';
line{36} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be
    removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <Tsrt);
[stp] = find(time(1,:) >Tstp);

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(1,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage2 = raw_data.variable_mat(12,:);
Vstage2(:,col) = [];
Vstage2(:,[strt stp]) = [];

Vedc = Vinput - Vstage2;

Pedc = raw_data.variable_mat(13,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

Eedc = raw_data.variable_mat(15,:);
Eedc(:,col) = [];
```

```
Eedc(:,[strt stp]) = [];

time = time-Tsrt; %sets the data to start at 0
time = time*1e6; %defines the time scale to be micro-seconds
Pedc = Pedc/1e3; %defines the power scale to be kilo-watts
Eedc = Eedc*1e3; %defines the energy scale to be mili-joules

EDC_bi_mat(:,1) = time';
EDC_bi_mat(:,2) = Vedc';
EDC_bi_mat(:,3) = Pedc';
EDC_bi_mat(:,4) = Eedc'-min(Eedc);

% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,2))
% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,3))
% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,4))

%% Data Save
save([data_filepath '\EDC_AK10530C_temp.mat'],'EDC_bi_mat');
```

Software C.34: TVS Diode Bidirectional Performance Values of TVS 3

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\SSCB_bi_Wolf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\TVS\littelfuse_tvs_diode_ak3_xxxc_b_spice_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_bi_Wolf_AK3380C']; %circuit simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\littelfuse_tvs_diode_ak3_xxxc_b_spice_temp.lib']; %EDC file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step

Tsrt = 0.9999999; %start time for data collection
Tstp = 1.000015; %stop time for data collection

Vin_vec = 1000; %input voltage

```

```
VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 2.5; %settings from datasheet

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                    {Vin
    }';
line{03} = 'XU1           input     gate_U1   TIE       Tj     Tc
    C2M0080120D';
line{04} = 'Rgate_U1      gate_U1   Vg1                                  {
    Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{06} = 'XU2           output    gate_U2   TIE       Tj     Tc
    C2M0080120D';
line{07} = 'Rgate_U2      gate_U2   VG2                                  {
    Rgate}';
line{08} = 'Vgate_U2      VG2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{09} = 'Lload         output    Load                                 {
    Lload}';
line{10} = 'Rload         Load      0                                    {
    Rload}';
line{11} = 'VTj           Tj        0                                    {VTj
    }';
line{12} = 'VTc           Tc        0                                    {VTc
    }';

line{13} = 'XU3             input  Stage1                          AK3-380C';
line{14} = 'XU4             Stage1 Stage2                          AK3-380C';
line{15} = 'XU5             Stage2 Stage3                          AK3-380C';
line{16} = 'Redc            Stage3 output                          1n';

line{17} = 'B1              Pedc   0                              V=(V(input)
    -V(Stage3))*I(Redc)';
line{18} = 'Ledc            Pedc   0                              1';

line{19} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
     ' ' num2str(Tmaxstep) ' startup'];
line{20} = ['.lib ' component_filepath];
line{21} = ['.lib ' EDC_filepath];
line{22} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
```

```
line{23} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg1)={VgI} V(Vg2)={VgI}';
line{24} = ['.param Vin = ' num2str(Vin_vec)];
line{25} = ['.param VgI = ' num2str(VgateI_vec)];
line{26} = ['.param VgF = ' num2str(VgateF_vec)];
line{27} = ['.param VgD = ' num2str(VgateD_vec)];
line{28} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{29} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{30} = ['.param VgO = ' num2str(VgateO_vec)];
line{31} = ['.param Rgate = ' num2str(Rgate_vec)];
line{32} = ['.param Rload = ' num2str(Rload_vec)];
line{33} = ['.param Lload = ' Lload_vec];
line{34} = ['.param VTj = ' num2str(Tj_vec)];
line{35} = ['.param VTc = ' num2str(Tc_vec)];
line{36} = '.backanno';
line{37} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "' circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. > In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <Tsrt);
[stp] = find(time(1,:) >Tstp);

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(1,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage3 = raw_data.variable_mat(13,:);
Vstage3(:,col) = [];
Vstage3(:,[strt stp]) = [];

Vedc = Vinput - Vstage3;

Pedc = raw_data.variable_mat(14,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];
```

```

Eedc = raw_data.variable_mat(16,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];

time = time-Tsrt; %sets the data to start at 0
time = time*1e6; %defines the time scale to be micro-seconds
Pedc = Pedc/1e3; %defines the power scale to be kilo-watts
Eedc = Eedc*1e3; %defines the energy scale to be mili-joules

EDC_bi_mat(:,1) = time';
EDC_bi_mat(:,2) = Vedc';
EDC_bi_mat(:,3) = Pedc';
EDC_bi_mat(:,4) = Eedc'-min(Eedc);

% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,2))
% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,3))
% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,4))

%% Data Save
save([data_filepath '\EDC_AK3380C_temp.mat'],'EDC_bi_mat');
```

Software C.35: TVS Diode Bidirectional Performance Values of TVS 4

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\SSCB_bi_Wolf.cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB\TVS\littelfuse_tvs_diode_ak6_xxxc_b_spice_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_bi_Wolf_AK6170C']; %circuit simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\littelfuse_tvs_diode_ak6_xxxc_b_spice_temp.lib']; %EDC file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step

Tsrt = 0.9999999; %start time for data collection
Tstp = 1.000015; %stop time for data collection
```

```
27
28 Vin_vec = 1000; %input voltage
29
30 VgateI_vec = -5; %initial gate voltage
31 VgateF_vec = 20; %final gate voltage
32 VgateD_vec = 0.5; %gate voltage delay
33 VgateTr_vec = 22 *1e-9; %rise time
34 VgateTf_vec = 14 *1e-9; %fall time
35 VgateO_vec = 0.5; %gate voltage ON time
36
37 Rgate_vec = 2.5; %settings from datasheet
38
39 Rload_vec = 30; %sets load resistance row vector
40 Lload_vec = '100u'; %sets load inductance row vector
41
42 Tj_vec = 25; %sets Tj row vector
43 Tc_vec = 25; %sets Tc row vector
44
45 %% MatLab Netlist
46 line{01} = ['* ' circuit_filepath '.asc'];
47 %BiDirectional Circuit
48 %           name          node1     node2     node3     node4  node5
    value / file with values
49 line{02} = 'Vin           input     0                                    {Vin
    }';
50 line{03} = 'XU1           input     gate_U1   TIE       Tj     Tc
    C2M0080120D';
51 line{04} = 'Rgate_U1      gate_U1   Vg1                                  {
    Rgate}';
52 line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
53 line{06} = 'XU2           output    gate_U2   TIE       Tj     Tc
    C2M0080120D';
54 line{07} = 'Rgate_U2      gate_U2   VG2                                  {
    Rgate}';
55 line{08} = 'Vgate_U2      VG2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
56 line{09} = 'Lload         output    Load                                 {
    Lload}';
57 line{10} = 'Rload         Load      0                                    {
    Rload}';
58 line{11} = 'VTj           Tj        0                                    {VTj
    }';
59 line{12} = 'VTc           Tc        0                                    {VTc
    }';
60
61 line{13} = 'XU3             input  Stage1                        AK6-170C';
62 line{14} = 'XU4             Stage1 Stage2                        AK6-170C';
63 line{15} = 'XU5             Stage2 Stage3                        AK6-170C';
64 line{16} = 'XU6             Stage3 Stage4                        AK6-170C';
65 line{17} = 'XU7             Stage4 Stage5                        AK6-170C';
66 line{18} = 'XU8             Stage5 Stage6                        AK6-170C';
67 line{19} = 'Redc            Stage6 output                        1n';
68
69 line{20} = 'B1              Pedc   0                             V=(V(input)
    -V(Stage6))*I(Redc)';
70 line{21} = 'Ledc            Pedc   0                             1';
71
```

```
line{22} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart) ' ' num2str(Tmaxstep) ' startup'];
line{23} = ['.lib ' component_filepath];
line{24} = ['.lib ' EDC_filepath];
line{25} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e-4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{26} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg1)={VgI} V(Vg2)={VgI}';
line{27} = ['.param Vin = ' num2str(Vin_vec)];
line{28} = ['.param VgI = ' num2str(VgateI_vec)];
line{29} = ['.param VgF = ' num2str(VgateF_vec)];
line{30} = ['.param VgD = ' num2str(VgateD_vec)];
line{31} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{32} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{33} = ['.param VgO = ' num2str(VgateO_vec)];
line{34} = ['.param Rgate = ' num2str(Rgate_vec)];
line{35} = ['.param Rload = ' num2str(Rload_vec)];
line{36} = ['.param Lload = ' Lload_vec];
line{37} = ['.param VTj = ' num2str(Tj_vec)];
line{38} = ['.param VTc = ' num2str(Tc_vec)];
line{39} = '.backanno';
line{40} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "' circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. > In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <Tsrt);
[stp] = find(time(1,:) >Tstp);

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(1,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage6 = raw_data.variable_mat(16,:);
Vstage6(:,col) = [];
Vstage6(:,[strt stp]) = [];
```

```
124
125 Vedc = Vinput - Vstage6;
126
127 Pedc = raw_data.variable_mat(17,:);
128 Pedc(:,col) = [];
129 Pedc(:,[strt stp]) = [];
130
131 Eedc = raw_data.variable_mat(19,:);
132 Eedc(:,col) = [];
133 Eedc(:,[strt stp]) = [];
134
135 time = time-Tsrt; %sets the data to start at 0
136 time = time*1e6; %defines the time scale to be micro-seconds
137 Pedc = Pedc/1e3; %defines the power scale to be kilo-watts
138 Eedc = Eedc*1e3; %defines the energy scale to be mili-joules
139
140 EDC_bi_mat(:,1) = time';
141 EDC_bi_mat(:,2) = Vedc';
142 EDC_bi_mat(:,3) = Pedc';
143 EDC_bi_mat(:,4) = Eedc'-min(Eedc);
144
145 % plot(EDC_bi_mat(:,1),EDC_bi_mat(:,2))
146 % plot(EDC_bi_mat(:,1),EDC_bi_mat(:,3))
147 % plot(EDC_bi_mat(:,1),EDC_bi_mat(:,4))
148
149 %% Data Save
150 save([data_filepath '\EDC_AK6170C_temp.mat'],'EDC_bi_mat');
```

Software C.36: TVS Diode Bidirectional Performance Values of TVS 5

```
1  %% TVS Performance
2  clc
3  close all
4  clear variables
5
6  %% Paths
7  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\SSCB_bi_Wolf
       .cir
8  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
9  %C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
       Masters\SSCB\TVS\littelfuse_tvs_diode_ak6_xxxc_b_spice_temp.lib
10 %C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe
11
12 %% Setup
13 parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
       Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
14 data_filepath = [parent_filepath '\TVS\']; %data file path
15 circuit_filepath = [data_filepath 'SSCB_bi_Wolf_AK6380C']; %circuit
       simulation file path
16 component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.
       lib']; %switch model file path
17 EDC_filepath = [parent_filepath '\TVS\
       littelfuse_tvs_diode_ak6_xxxc_b_spice_temp.lib']; %EDC file path
18
19 %all parameters are scriptable, use .step param or .step param XXX list
20 Tprint = 0;
```

```
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step

Tsrt = 0.9999999; %start time for data collection
Tstp = 1.000015; %stop time for data collection

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 2.5; %settings from datasheet

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                  {Vin
    }';
line{03} = 'XU1           input     gate_U1   TIE       Tj     Tc
    C2M0080120D';
line{04} = 'Rgate_U1      gate_U1   Vg1                                {
    Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{06} = 'XU2           output    gate_U2   TIE       Tj     Tc
    C2M0080120D';
line{07} = 'Rgate_U2      gate_U2   VG2                                {
    Rgate}';
line{08} = 'Vgate_U2      VG2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{09} = 'Lload         output    Load                               {
    Lload}';
line{10} = 'Rload         Load      0                                  {
    Rload}';
line{11} = 'VTj           Tj        0                                  {VTj
    }';
line{12} = 'VTc           Tc        0                                  {VTc
    }';

line{13} = 'XU3             input  Stage1                          AK6-380C';
line{14} = 'XU4             Stage1 Stage2                          AK6-380C';
line{15} = 'XU5             Stage2 Stage3                          AK6-380C';
line{16} = 'Redc            Stage3 output                          1n';

line{17} = 'B1              Pedc   0                              V=(V(input)
    -V(Stage3))*I(Redc)';
```

```
line{18} = 'Ledc            Pedc   0                                1';

line{19} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart)
    ' ' num2str(Tmaxstep) ' startup'];
line{20} = ['.lib ' component_filepath];
line{21} = ['.lib ' EDC_filepath];
line{22} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e
    -4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{23} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg1)={VgI} V(Vg2)={
    VgI}';
line{24} = ['.param Vin = ' num2str(Vin_vec)];
line{25} = ['.param VgI = ' num2str(VgateI_vec)];
line{26} = ['.param VgF = ' num2str(VgateF_vec)];
line{27} = ['.param VgD = ' num2str(VgateD_vec)];
line{28} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{29} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{30} = ['.param VgO = ' num2str(VgateO_vec)];
line{31} = ['.param Rgate = ' num2str(Rgate_vec)];
line{32} = ['.param Rload = ' num2str(Rload_vec)];
line{33} = ['.param Lload = ' Lload_vec];
line{34} = ['.param VTj = ' num2str(Tj_vec)];
line{35} = ['.param VTc = ' num2str(Tc_vec)];
line{36} = '.backanno';
line{37} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "'
    circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The
    encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. >
     In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be
    removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <Tsrt);
[stp] = find(time(1,:) >Tstp);

time(:,[strt stp]) = [];

Vinput = raw_data.variable_mat(1,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage3 = raw_data.variable_mat(13,:);
```

```
Vstage3(:,col) = [];
Vstage3(:,[strt stp]) = [];

Vedc = Vinput - Vstage3;

Pedc = raw_data.variable_mat(14,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

Eedc = raw_data.variable_mat(16,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];

time = time-Tsrt; %sets the data to start at 0
time = time*1e6; %defines the time scale to be micro-seconds
Pedc = Pedc/1e3; %defines the power scale to be kilo-watts
Eedc = Eedc*1e3; %defines the energy scale to be mili-joules

EDC_bi_mat(:,1) = time';
EDC_bi_mat(:,2) = Vedc';
EDC_bi_mat(:,3) = Pedc';
EDC_bi_mat(:,4) = Eedc'-min(Eedc);

% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,2))
% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,3))
% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,4))

%% Data Save
save([data_filepath '\EDC_AK6380C_temp.mat'],'EDC_bi_mat');
```

Software C.37: TVS Diode Bidirectional Performance Values of TVS 6

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\SSCB_bi_Wolf
    .cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\TVS\PTVS10-xxxC-TH_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
circuit_filepath = [data_filepath 'SSCB_bi_Wolf_PTVS10']; %circuit
    simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.
    lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\PTVS10-xxxC-TH_temp.lib']; %EDC file
    path

```

```
%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step

Tsrt = 0.9999999; %start time for data collection
Tstp = 1.000015; %stop time for data collection

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 2.5; %settings from datasheet

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5
    value / file with values
line{02} = 'Vin           input     0                                {Vin
    }';
line{03} = 'XU1           input     gate_U1   TIE       Tj     Tc
    C2M0080120D';
line{04} = 'Rgate_U1      gate_U1   Vg1                              {
    Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{06} = 'XU2           output    gate_U2   TIE       Tj     Tc
    C2M0080120D';
line{07} = 'Rgate_U2      gate_U2   VG2                              {
    Rgate}';
line{08} = 'Vgate_U2      VG2       TIE
    PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{09} = 'Lload         output    Load                             {
    Lload}';
line{10} = 'Rload         Load      0                                {
    Rload}';
line{11} = 'VTj           Tj        0                                {VTj
    }';
line{12} = 'VTc           Tc        0                                {VTc
    }';

line{13} = 'XU3           input  Stage1
    PTVS10-380C-TH';
line{14} = 'XU4           Stage1 Stage2
    PTVS10-380C-TH';
```

```
line{15} = 'XU5            Stage2 Stage3                                                PTVS10-380C-TH';
line{16} = 'Redc           Stage3 output                                                1n';

line{17} = 'B1             Pedc   0                                                     V=(V(input)-V(Stage3))*I(Redc)';
line{18} = 'Ledc           Pedc   0                                                     1';

line{19} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart) ' ' num2str(Tmaxstep) ' startup'];
line{20} = ['.lib ' component_filepath];
line{21} = ['.lib ' EDC_filepath];
line{22} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e-4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{23} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg1)={VgI} V(Vg2)={VgI}';
line{24} = ['.param Vin = ' num2str(Vin_vec)];
line{25} = ['.param VgI = ' num2str(VgateI_vec)];
line{26} = ['.param VgF = ' num2str(VgateF_vec)];
line{27} = ['.param VgD = ' num2str(VgateD_vec)];
line{28} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{29} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{30} = ['.param VgO = ' num2str(VgateO_vec)];
line{31} = ['.param Rgate = ' num2str(Rgate_vec)];
line{32} = ['.param Rload = ' num2str(Rload_vec)];
line{33} = ['.param Lload = ' Lload_vec];
line{34} = ['.param VTj = ' num2str(Tj_vec)];
line{35} = ['.param VTc = ' num2str(Tc_vec)];
line{36} = '.backanno';
line{37} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "' circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. > In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be removed

time = raw_data.time_vect;
time(:,col) = [];

[strt] = find(time(1,:) <Tsrt);
[stp] = find(time(1,:) >Tstp);

time(:,[strt stp]) = [];
```

```

Vinput = raw_data.variable_mat(1,:);
Vinput(:,col) = [];
Vinput(:,[strt stp]) = [];

Vstage3 = raw_data.variable_mat(13,:);
Vstage3(:,col) = [];
Vstage3(:,[strt stp]) = [];

Vedc = Vinput - Vstage3;

Pedc = raw_data.variable_mat(14,:);
Pedc(:,col) = [];
Pedc(:,[strt stp]) = [];

Eedc = raw_data.variable_mat(16,:);
Eedc(:,col) = [];
Eedc(:,[strt stp]) = [];

time = time-Tsrt; %sets the data to start at 0
time = time*1e6; %defines the time scale to be micro-seconds
Pedc = Pedc/1e3; %defines the power scale to be kilo-watts
Eedc = Eedc*1e3; %defines the energy scale to be mili-joules

EDC_bi_mat(:,1) = time';
EDC_bi_mat(:,2) = Vedc';
EDC_bi_mat(:,3) = Pedc';
EDC_bi_mat(:,4) = Eedc';

% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,2))
% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,3))
% plot(EDC_bi_mat(:,1),EDC_bi_mat(:,4))

%% Data Save
save([data_filepath '\EDC_PTVS10_temp.mat'],'EDC_bi_mat');
```

Software C.38: TVS Diode Bidirectional Performance Values of TVS 7

```
%% TVS Performance
clc
close all
clear variables

%% Paths
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D\EnergyDissipation\SSCB_bi_Wolf
    .cir
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\MOSFET\Wolfspeed\C2M0080120D_temp.lib
%C:\Users\2677hughesm\Marquette University\Power Electronics Lab - HughesMD\
    Masters\SSCB\TVS\PTVS3-xxxC-TH_HV_temp.lib
%C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe

%% Setup
parent_filepath = 'C:\Users\2677hughesm\Marquette University\Power
    Electronics Lab - HughesMD\Masters\SSCB'; %parent file path
data_filepath = [parent_filepath '\TVS\']; %data file path
```

```
circuit_filepath = [data_filepath 'SSCB_bi_Wolf_PTVS3']; %circuit simulation file path
component_filepath = [parent_filepath '\MOSFET\Wolfspeed\C2M0080120D_temp.lib']; %switch model file path
EDC_filepath = [parent_filepath '\TVS\PTVS3-xxxC-TH_HV_temp.lib']; %EDC file path

%all parameters are scriptable, use .step param or .step param XXX list
Tprint = 0;
Tstop = 1.5; %transient analysis stop time
Tstart = 0;
Tmaxstep = 1 *1e-3; %transient analysis step

Tsrt = 0.9999999; %start time for data collection
Tstp = 1.00002; %stop time for data collection

Vin_vec = 1000; %input voltage

VgateI_vec = -5; %initial gate voltage
VgateF_vec = 20; %final gate voltage
VgateD_vec = 0.5; %gate voltage delay
VgateTr_vec = 22 *1e-9; %rise time
VgateTf_vec = 14 *1e-9; %fall time
VgateO_vec = 0.5; %gate voltage ON time

Rgate_vec = 2.5; %settings from datasheet

Rload_vec = 30; %sets load resistance row vector
Lload_vec = '100u'; %sets load inductance row vector

Tj_vec = 25; %sets Tj row vector
Tc_vec = 25; %sets Tc row vector

%% MatLab Netlist
line{01} = ['* ' circuit_filepath '.asc'];
%BiDirectional Circuit
%           name          node1     node2     node3     node4  node5          value / file with values
line{02} = 'Vin           input     0                                      {Vin}';
line{03} = 'XU1           input     gate_U1   TIE       Tj     Tc           C2M0080120D';
line{04} = 'Rgate_U1      gate_U1   Vg1                                    {Rgate}';
line{05} = 'Vgate_U1      Vg1       TIE       PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{06} = 'XU2           output    gate_U2   TIE       Tj     Tc           C2M0080120D';
line{07} = 'Rgate_U2      gate_U2   VG2                                    {Rgate}';
line{08} = 'Vgate_U2      VG2       TIE       PULSE({VgI} {VgF} {VgD} {VgTr} {VgTf} {VgO})'; %settings from datasheet
line{09} = 'Lload         output    Load                                   {Lload}';
line{10} = 'Rload         Load      0                                      {Rload}';
line{11} = 'VTj           Tj        0                                      {VTj}';
```

```
line{12} = 'VTc            Tc         0                                        {VTc}';

line{13} = 'XU3            input   Stage1                                      PTVS3-380C-TH';
line{14} = 'XU4            Stage1  Stage2                                      PTVS3-380C-TH';
line{15} = 'XU5            Stage2  Stage3                                      PTVS3-380C-TH';
line{16} = 'Redc           Stage3  output                                      1n';

line{17} = 'B1             Pedc    0                                           V=(V(input)-V(Stage3))*I(Redc)';
line{18} = 'Ledc           Pedc    0                                           1';

line{19} = ['.tran ' num2str(Tprint) ' ' num2str(Tstop)  ' ' num2str(Tstart) ' ' num2str(Tmaxstep) ' startup'];
line{20} = ['.lib ' component_filepath];
line{21} = ['.lib ' EDC_filepath];
line{22} = '.options METHOD=GEAR GMIN=1e-7 ABSTOL=1e-6 RELTOL=1e-2 VNTOL=1e-4 CHGTOL=1e-10 ITL1=200 ITL2=100 ITL4=20 ITL6=50';
line{23} = '.ic V(input)={Vin} V(Tj)={VTj} V(Tc)={VTc} V(Vg1)={VgI} V(Vg2)={VgI}';
line{24} = ['.param Vin = ' num2str(Vin_vec)];
line{25} = ['.param VgI = ' num2str(VgateI_vec)];
line{26} = ['.param VgF = ' num2str(VgateF_vec)];
line{27} = ['.param VgD = ' num2str(VgateD_vec)];
line{28} = ['.param VgTr = ' num2str(VgateTr_vec)];
line{29} = ['.param VgTf = ' num2str(VgateTf_vec)];
line{30} = ['.param VgO = ' num2str(VgateO_vec)];
line{31} = ['.param Rgate = ' num2str(Rgate_vec)];
line{32} = ['.param Rload = ' num2str(Rload_vec)];
line{33} = ['.param Lload = ' Lload_vec];
line{34} = ['.param VTj = ' num2str(Tj_vec)];
line{35} = ['.param VTc = ' num2str(Tc_vec)];
line{36} = '.backanno';
line{37} = '.end';

%% New Netlist
fid = fopen([circuit_filepath '.cir'],'wb');
for i = 1:length(line)
    fwrite(fid, [line{i} char(13) newline], 'char');
end
fid = fclose(fid);

%% Simulate
system(['"C:\Program Files\LTC\LTspiceXVII\XVIIx64.exe" -b "' circuit_filepath '.cir"']);

%% Data Collection
warning('off', 'MATLAB:iofun:UnsupportedEncoding'); %Removes Warning: The encoding 'UTF-16LE' is not supported. See the documentation for FOPEN. > In LTspice2Matlab (line 153) In SSCB_efficiency (line 68)
raw_data = LTspice2Matlab([circuit_filepath '.raw']);

%BiDirectional Circuit
[col] = find(isnan(raw_data.variable_mat(1,:))); %finds NAN columns to be removed

```

```
106 time = raw_data.time_vect;
107 time(:,col) = [];
108
109 [strt] = find(time(1,:) <Tsrt);
110 [stp] = find(time(1,:) >Tstp);
111
112 time(:,[strt stp]) = [];
113
114 Vinput = raw_data.variable_mat(1,:);
115 Vinput(:,col) = [];
116 Vinput(:,[strt stp]) = [];
117
118 Vstage3 = raw_data.variable_mat(13,:);
119 Vstage3(:,col) = [];
120 Vstage3(:,[strt stp]) = [];
121
122 Vedc = Vinput - Vstage3;
123
124 Pedc = raw_data.variable_mat(14,:);
125 Pedc(:,col) = [];
126 Pedc(:,[strt stp]) = [];
127
128 Eedc = raw_data.variable_mat(16,:);
129 Eedc(:,col) = [];
130 Eedc(:,[strt stp]) = [];
131
132 time = time-Tsrt; %sets the data to start at 0
133 time = time*1e6; %defines the time scale to be micro-seconds
134 Pedc = Pedc/1e3; %defines the power scale to be kilo-watts
135 Eedc = Eedc*1e3; %defines the energy scale to be mili-joules
136
137 EDC_bi_mat(:,1) = time';
138 EDC_bi_mat(:,2) = Vedc';
139 EDC_bi_mat(:,3) = Pedc';
140 EDC_bi_mat(:,4) = Eedc';
141
142 % plot(EDC_bi_mat(:,1),EDC_bi_mat(:,2))
143 % plot(EDC_bi_mat(:,1),EDC_bi_mat(:,3))
144 % plot(EDC_bi_mat(:,1),EDC_bi_mat(:,4))
145
146 %% Data Save
147 save([data_filepath '\EDC_PTVS3_temp.mat'],'EDC_bi_mat');
```

Software C.39: TVS Diode Bidirectional Performance Values of TVS 8